PERSON-Personalised Expert Recommendation System for Optimised Nutrition

Chih-Han Chen

Department of Electrical and Electronic Engineering
Imperial College London

This thesis is submitted for the degree of
Doctor of Philosophy

Imperial College London November 2019

Declaration

I hereby declare that this thesis is my own work. Except where appropriately references, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification, or any other university.

Chih-Han Chen
November 2019

Abstract

The Nobel-Prize-winning concept of nudge theory aims to optimise the behaviour of individuals through making small changes in daily life. In the food industry, research into nudge theory is moving towards the application of healthier decisions via personalisation.

Deoxyribonucleic acid (DNA) provides the best basis for research into personalisation, as it is known to be the key to understanding the development and functioning of all living organisms. DNA analysis services have emerged in the consumer market, offering people access to their DNA patterns and relevant information. However, due to the high complexity and large amount of information involved, existing knowledge in this field is rarely applied in daily life. In order to employ the knowledge gained from genetic innovations within a market context, research into a framework that can provide optimised decisions to consumers, retailers, and manufacturers is of interest. In this work, I develop a personalised expert recommendation system for optimised nutrition (PERSON). This can perform personalised grocery product filtering for consumers and generate recommendations based on correlations between nutrition and genetics. I also construct a simulation framework called grocery business unionism strategy in neural expert system simulation (GBUSiNESS), which simulates consumer agent models and manufacturer agent models utilising PERSON in order to generate better product productions and marketing strategies for suppliers. To confirm the advantages of my expert recommendation systems, a trial was conducted with customers in a supermarket.

The contribution of the invented framework PERSON is in generating recommendations that can form the basis for better product decisions for consumers, with simulation framework GBUSiNESS that can enable retailers and manufacturers to improve food manufacturing based on genetic knowledge. This project is an example of DNAnudge, and can demonstrate the application of nudge theory to recommendation systems for personalised food decisions. It is expected to benefit both customers and manufacturers.

Acknowledgements

This research project would not have been possible without the supervision of Professor Christofer Toumazou at Imperial College London and the support from DNAnudge Ltd. First and foremost, I would like to thank Prof. Toumazou for this opportunity and for his guidance, innovation and creativity. I would also like to thank Dr. Maria Karvela for her mentoring and Dr. Mohammadreza Sohbati for his inspiration. Furthermore I would like to express my gratitude to my colleagues at DNAnudge: by working together, we have converted theoretical results into applications. Finally I would like to thank my parents Shyh-Yong Chen, Yu-Han Tsou and my wife Qi Hao for their continuous and unstinting support for my studies and research.

Table of contents

List of Abbreviations and Symbols

Abbreviations

AI Artificial Intelligence

B2B Business to Business

B2C Business to Consumer

BoW Bag-of-Word

CNN Convolutional Neural Network

CPUs Central Processing Units

DEPTA Data Extraction based on Partial Tree Alignment

DLSTM Deep Long Short-Term Memory

DNA Deoxyribonucleic

DNNs Deep Neural Networks

DOM Document Object Model

DPM Data Path Matching

DTC Direct To Consumer

FPGA Field Programmable Gate Arrays

G-BUSiNESS Grocery Business Unionism Strategy in Neural Expert System Simulation

GA Genetic Algorithm

GPUs Graphics Processing Units

HTML Hypertext Transfer Protocol

HTTP Hypertext Transfer Protocol

IEEE Institute of Electrical and Electronics Engineers

IEPAD Information Extraction based on Pattern Discover

LSTM Long Short-Term Memory

MDP Markov Decision Processe

mRNA Messenger Ribonucleic Acid

NLP Natural Language Processing

PERSON Personalised Expert Recommendation System for Optimised Nutrition

PERSON Personalised Expert Recommendation System for Optimised Nutrition

PHP Hypertext Preprocessor

RNN Recurrent Neural Network

SDdiff Variation of the Standard Deviation

SNP Single Nucleotide Polymorphism

SQL Structured Query Language

SVM Support Vector Machine

t-SNE t-distributed Stochastic Neighbour Embedding

URL Uniform Resource Identifier

USDA United States Department of Agriculture

Symbols of the framework PERSON

η	Learning rate
λ	Margin size
ϕ	Projection kernel function
σ	Activation function for LSTM
a	Average nutritional value
b	Bias of the model
c	Coefficients for the SVM model
D	Decision of the suggesting function
E	Environmental factor
e	Learning signal
fi	Fitness function of genetic algorithm
fo	Forget gate of LSTM model
f	Activation function
G	Genotype
g	Input state of LSTM model
H	Low pass filtering function
h	Hidden layer of a neuron network
in	Input gate of LSTM model
i	Horizontal identifier of an input data
j	Vertical identifier of an input data
k	Identifier of a weight location
Len	String length of genetic algorithm
l	Level of filter of PERSON

n Identifier of nutritional factor

ou Output gate of LSTM model

Pc Probability of crossover of genetic algorithm

Pm Probability of mutation of genetic algorithm

Po Population size of genetic algorithm

Pre Distribution of the model prediction

P Phenotype

p Product nutrients features

Q Probability of following recommendation of a consumer

r Reduced consumption nutrient

s Variable representing scanned result

Te Termination criteria of genetic algorithm

T Threshold table of PERSON

u Processing layer of the model

v Weighted nutritional value

W Weight of the model

X Variable representing input

y Output of the model

Symbols of the Case Study

F Forget parameter of the supermarket trial

g Group identifier of consumer

MP Monthly personal effect of the supermarket trial

M Monthly bias between consumers

m Month

P Personal bias between consumers

RD DNAnudge effect of reference group of the supermarket trial

RS Self-nudge effect of reference group of the supermarket trial

R Reference group of the supermarket trial

TD DNAnudge effect of target group of the supermarket trial

TI Immediate DNA Nudge effect of target group of the supermarket trial

TM Memory DNAnudge effect of target group of the supermarket trial

TS Self-nudge effect of target group of the supermarket trial

T Target group of the supermarket trial

U Recorded green product purchase of the supermarket trial

W Slope parameter of a consumer group

y Year

Symbols of the framework GBUSiNESS

γ Discount rate

π Policy map

a Action

b Benefit boundary

d Difference between a product feature and boundary

E Expected return

f Product feature type

g Number of features involved with taste

h Hidden layer of a neuron network

i Product identifier

L Loss function

MU Modification unit

NF Nutritional feature list

NW Weight of a neuron network

n Nutritional feature

o Observation of manufacturer agent

PF Product feature list

PI Profit improvement

PL Product identifier list

P Probability of transition

p Product feature vector ordered by product identifier

Q Action value function

q Product feature vector ordered by product feature types

R Cumulative reward

r Reward

SI Sales improvement

s State

t Time step

V Value function

W Weight of a type of product feature

y Output label of a neuron network

z State, action and reward set

θ Weight for state and action pair

Chapter 1

Introduction

"Nudge" theory is a game-changing concept that has been applied by governments, large corporations, and even small retailers. It aims to modify the behaviour of individuals by making small changes in daily life. The theory grew significantly in influence and popularity when Richard Thaler won the Nobel Prize for Economics for its development in 2017. Among the numerous studies and applications that now exist in economics, one of the key areas in which the nudge concept can be applied is health care, which has attracted increasing attention in recent years. The revolutionary changes in this area have involved the personalisation of health care treatments, such as personalised diets, disease prevention, and skin care. In the field of dietary advice, nutrigenomics and nutrigenetics have been developed to characterise the effects of nutrients on interactions among genes, proteomes, and metabolomes. As the saying goes, "You are what you eat". In this field, it is believed that all of the molecules involved in human metabolism are controlled by the language of genes, which can be controlled and radically alter how food interacts with the human body in order to optimises health [57, 101]. However, the complexity of the available information has limited the practical applications of this approach, stimulating research into various systems that can provide personalised dietary advice.

To deal with the complexity of this information and to bridge various forms of knowledge towards the development of an application, I propose an expert system that aims to offer personalised recommendations based on genetics. Using this expert system, I present the first example of a concept called "DNAnudge". DNAnudge is an extension of nudge theory and is designed to "nudge" people towards better decisions based on their genetic patterns. My research makes three main contributions: an application that performs personalised recommendations, a trial that demonstrates the concept's application; and a simulation that predicts the behaviours of consumers and grocery manufacturers.

The most realistic way of demonstrating the influence of a concept is through an application, for example an expert system. Since 1998, expert systems have been accepted as a solution for

higher production quality and better dietary choices in the food industry [75]. Together with the advancement of big data and cloud technology, systems that can cope with a large variety, volume and velocity of data can bring success to business [79], science and health care [52]. Indeed, a recommendation system that can scale to handle big data through MapReduce has led to the creation of numerous applications, including product [74], music and health recommendation systems [56].

In the design of any system, scalability is an important factor. The scalability of models within all systems requires an ability to deal with unknown data [1], which is usually measured by the generality of models. In the area of health care, the introduction of deep learning analysis - and particularly deep neural networks (DNNs) - is believed to be the reason why recommendation expert systems have improved in recent years [56]. Neural networks, which aim to simulate the behaviour of neurons, started by utilising existing devices, and following the development of programming frameworks, evolved into circuit implementation. Technology companies such as Microsoft and Google have published open-source research tools that have the ability to effectively simulate neural networks using the computes' central processing units (CPUs) or graphics processing units (GPUs). These tools allow researchers to focus more closely on the design of network structures and applications [117, 114]. Due to the need for greater accuracy and better performances when solving problems that arise from scaling in terms of neuron sizes, research into neural network systems has expanded into FPGA programming and in the future it is expected to be fully implemented in circuits down to the device level [114].

From an application perspective, personalised health care and diets have received considerable attention since the rise of nutri-genetics. Direct-to-consumer (DTC) genetic services have emerged [12] following the invention of cost-effective genetic analysis tests [131], which have led to high expectations for associated applications. One survey has shown that up to 50% of subjects are willing to undergo genetic testing and to follow personalised suggestions for behavioural and lifestyle changes [23]. However, the information that most of these services provide to consumers is complicated, resulting in difficulties in understanding them, let along following them to keep a personalised diet.

In order to provide meaningful and straight-forward solutions, a novel expert system is proposed here that aims to personalise grocery shopping through product suggestions based on genetic phenotypes. Existing studies on systems for dietary decisions and personalisation do not consider genetic phenotypes. For example, one system that seeks to provide decision support for food choices focuses only on the temporal aspects of a diet using optimised reasoning, rather than using genetics as its input [7]. Another similar application is a so-called medical

education system that provides personalised suggestions based on individual characteristics and health objectives [104].

In this work, I focus on providing consumers with temporal recommendations based on genetics, including comparisons between particular grocery products and all other products within the same food category. I also aim to achieve personalised recommendations based on the nutritional information of grocery products. A system called PERSON (Personalised Expert Recommendation System for Optimised Nutrition) is constructed. The system is designed to process two data sets: genetic data and grocery data. Genetic data relating to phenotypes can influence the consumption ability of five primary nutritional factors: energy, fat, protein, sugar, and salt. These are used to perform simulations of all possible combinations. The genetic data collected from customers are assumed to be static, i.e. to remain the same over time. On the other hand, grocery data is designed to be dynamic and to vary with continuous data collection. In this thesis, I describe in detail how the grocery data is processed, how the genetic data is used for recommendations and how the whole system is put together as a single application, built upon invented models that perform categorisations and recommendations. I have published this work on the applications of DNAnudge in the areas of nutrition and diet in IEEE Transactions on Biomedical Circuits and Systems, in a paper titled "PERSON-Personalised Expert Recommendation System for Optimised Nutrition" [22]. A report on the same system is also published as a chapter in the book *Trends in Personalized Nutrition*, published by Elsevier on 24 May 2019 [44].

I then construct a framework consisting of consumer agent models and manufacturer agent models (trained by reinforcement learning) to simulate the behaviours of a business union and to compare the influences of PERSON. The consumer agent model can simulate **the** behaviours of different consumers with various genes and compare them between personalised diets and generalised diets. Consumers strive to optimise their decisions on products while considering factors such as price, nutrition, and taste. The manufacturer agent model can simulate the behaviours of different manufacturers with various products. Manufacturers aim to maximise their profit through modifying their product features, including price cost and nutrition. The modification of products can affect consumer decisions, which causes rewards or punishments for manufacturers. In a business union, manufacturer agents compete with each other following different strategies by modifying their product features. This simulation framework can help manufacturers to optimise their product features and can confirm the benefits of a personalised diet based on genes.

Finally, Working with a DNAnudge behavioural science team and a supermarket, I constructed a trial in a supermarket with real-world consumers. PERSON was implemented as a mobile service application called DNAnudge. By analysing the real-world data, I demonstrated

the two novel terms defined by professor Christofer Toumazou: a self-controlled decision variation called "Self-nudge" and a DNA-driven decision variation called "DNAnudge". **Self-nudge** aims to explain the variations of individual behaviours without any influence from outside. **DNAnudge** is a nudge process targeted towards changing individual behaviours based on their genes (recommendations from PERSON). I developed a mathematical model to represent Self-nudge and DNAnudge based on real-world data. This model and trial is my fourth contribution in this research that further confirms the advantages of my proposed PERSON expert system.

This research makes the following three primary technical contributions:

1. An application called PERSON consisting of a categorisation model and a recommendation model introduced in chapter 4

2. A trial and influence demonstration of PERSON through a mathematical model generated based on real-world data covered in chapter 5; and

3. A simulation model called GBUSiNESS bringing PERSON into the business environment by providing decision-making assistance for product manufacturers outlined in Chapter 6.

The remainder of this thesis is organised as follows. Chapter 2 contains a brief review of nudge theory, nutrigenetics and nutrigenomics. In Chapter 3, the approaches to data collection and processing are introduced and a generalised motivation and system framework for DNAnudge is described, along with a discussion of generalisation and scalability issues. In chapters 4, 5 and 6, I cover my three main contributions, starting with a categorisation and recommendation model application in chapter 4. The categorisation model is described in section 4.2, where I have built a machine-learning model with a word-embedding technique to categorise grocery products. In section 4.4, the recommendation model is introduced, and I explain how the genetic data is used and how the algorithm is applied to generate personalised recommendations for optimised nutrition. Chapter 5 contains a real-world supermarket trial with consumers. The mathematical model based on real-world data confirms the influences of Self-nudge and DNAnudge. Finally, in Chapter 6, the simulation model is described, and I present the concept of a simulation system as an extension of the PERSON expert system. This system seeks to simulate the behaviours of consumers and business unionism with game theory. I explain three experiments applied in the simulation system with consumer and manufacturer reinforcement-learning artificial intelligence (AI) agents. The thesis concludes with Chapter 7, where I discuss the completed studies. Chapter 7 also summarises the original contribution of this work and discusses possible extensions in its sections on future work and final thoughts.

Chapter 2

Nudge Theory, Nutrigenetics and Nutrigenomics

Nudge theory and nutrigenetics represent the two main foundations of this research project. This chapter covers some of the current progress in and background of the two studies. In Section 2.1, the concept and application of nudge are explained. The connection between nutrition and genetics is described in Section 2.2.

2.1 Nudge Theory

Nudge theory proposes that indirect suggestions influence the decisions and behaviours of individuals and groups. At present, research is actively being undertaken into this concept in behavioural science, political theory and behavioural economics. The idea differs from the view that compliance results from legislation. In business organisations, decision makers apply nudge theory in their management and corporate culture, particularly in areas related to health, safety, and the environment. For example, companies in Silicon Valley apply nudge theory in corporate settings to increase the productivity and happiness of their employees [38].

James Wilk formulated the formal concept of nudge, and in 1997 linked it to various principles in cybernetics [134]. At that time nudge was used in micro-targeted designs to influence a specific type of group, without considering the scale of the group. The book *Nudge: Improving Decisions About Health, Wealth, and Happiness*, which was written in 2008 by Richard Thaler and Cass Sunstein, brought the theory to its peak in the economic domain by considering the scale of the groups. The book stimulated growing number of discussions about 'nudge' in both the United States of America(USA) and the United Kingdom(UK) and within the private sectors and public health areas. There is particular interest in this theory in the

domains of psychology and behavioural economics, because it is concerned with influencing individuals' behaviour without being criticised and it has also been referred to as libertarian paternalism rather than acting as choice architects [121, 130]. Interest in nudge theory grew significantly when Richard Thaler won the Nobel Prize for Economics in 2017.

The key principle of nudge is to prompt individuals to behave in a particular way or to make a particular decision, through changing part of the environment to which they are exposed. This environmental factor influences the behaviour of the individuals through triggering automatic cognitive processes [111]. The term 'value action gap' describes the issue that the behaviours of an individual are not always aligned with his or her intentions. It is a widely known fact that humans are not constantly rational. People often carry out actions that are not in their interests, when they are aware of these actions [69]. A typical example is hungry dieters. A hungry dieter is very likely to ignore or temporarily weaken his or her plan to lose weight or to eat in a healthy way, and may regret these actions when they are satisfied with unhealthy food [95].

The reason for actions caused by the 'value action gap' can be explained by the thinking systems of human beings. Sunstein and Thaler proposed that thinking has two types of systems: the 'automatic system' and the 'reflective system' [71]. The automatic system is related to the instinctive and rapid approach to thinking; examples include the reaction of smiling at the sight of a cute animal or becoming nervous when in a high pressure environment. The reflective system is self-conscious; examples of this type of thinking are making a decision about where to work or where to go on holidays [71]. The combination of these two systems causes biases, heuristics, and fallacies, which are extended to five main topics in Sunstein and Thaler's book: anchoring, the availability heuristic, the representativeness heuristic, the status quo bias and the herd mentality.

- **Anchoring:** this is a cognitive bias caused by beliefs that heavily depend on one trait or one piece of information. Once the value of the anchor is set, the bias usually occurs when interpreting information in the future.

- **Availability heuristic:** this is applied when someone considers a probability based on the available examples in his or her mind. One example is the belief that homicides occur more often than suicides, which arises because examples of homicides are more likely to be available in one's mind. This bias causes people to overstate risks, and brings uncertainty when making decisions.

- **Representativeness heuristic:** this occurs when judgements and hypotheses are made on the basis of the available data. A good example is the extraction of meaningful patterns from information that is actually random.

- **Status quo bias:** this occurs when people continue with an action, even though the action is not in their interest. An example is a magazine subscription, which usually starts with a free period. After the free period has expired, the subscription continues until the users actively cancel the subscription and stops paying.

- **Herd mentality:** this reflects the fact that people's decisions are heavily influenced by other people. A study carried out by Solomon Asch, cited in Sunstein and Thaler's book, showed that people answer questions in a way that is clearly incorrect only because of peer pressures.

Libertarian paternalism is also an important concept and represents two political notions that have opposite meanings. Sunstein and Thaler emphasise that "the libertarian aspect of my strategies lies in the straightforward insistence that, in general, people should be free to do what they like-and to opt out of undesirable arrangements if they want to do so" and that "the term paternalistic lies in the claim that it is legitimate for choice architects to try to influence people's behaviour in order to make their lives longer, healthier, and better" [71].

Choice architecture is another important concept in nudge theory. It describes different approaches to influencing decisions, through changing the ways in which choices are presented. The aim is to nudge individuals through choice architectures, while affording them the ability to make their decisions freely. There are three main types of nudges: defaults, social proof heuristics, and modifying the salience of the desired option. A default option of nudge emphasises a particular option for people when it is automatically generated [17]. An example is to set a good health programme as the default choice for older people, aiming to decrease the number of older people who are outside any programme for health care. Meanwhile, drug comparisons are also reported, and are freely available and easily accessible [71]. A social proof heuristics describes a situation in which an individual's behaviours is guided by other people's behaviour. Some studies have shown success in using social proof heuristics nudges for healthier food decisions [24]. Nudging through modifying the salience of the desired option is done to increase or decrease the desire to make a particular choice. An example of such nudge architecture in a shop is placing healthier foods at eye level and less healthy junk food in hard-to-reach places. In this example, individuals are not prevented from eating the junk food, but due to differences in ease of access people tend to eat the healthier food [71]. Applications of nudge theory already exist in everyday life, such as in the work carried out by Sunstein as an administrator in the Office of Information and Regulatory Affairs in the United States in 2008 [121]. In 2010, nudge theory was included in the formation of the British Behavioural Insights Team. Furthermore, both the former Prime Minister of the UK, David Cameron and the

former President of the USA Barack Obama made efforts to employ nudge theory to improve domestic policy goals during their terms of office [81].

In this PhD research, I propose an application of nudge theory in the area of health and nutrition, targeting the achievement of better and healthier decisions regarding food choices based on genetics. The key components of nutrition and genetics are covered in the following section.

2.2 Nutrigenetics and Nutrigenomics

In nutrition research, studies into how nutrition optimises and maintains cellular, tissue, organ and whole body homeostasis have resulted in a shift in interests from epidemiology and physiology to nutrigenomics-molecular and nutrigenetics-molecular. These research studies require nutrients to act at the molecular level and to interact with genes and proteins. [93, 16, 68].

Nutrigenomics and nutrigenetics aim to unfold the role of nutrition and gene expression in order to understand the relationship between diet and health [93, 42]. The principle underlying both fields was first commercialised by a number of companies in the 1980s. By the 1990s the Human Genome Project which sequenced the entire DNA in human genomes further boosted the excitement of researchers [93]. In the 20th century, nutritional scientists focused on finding and defining the use of vitamins and minerals [46, 82]. As nutrition-related health problems such as obesity and type-2 diabetes became more severe by 2007, modern medicine and nutritional science have increasingly focused on how to prevent these diseases [105]. The key difference between nutrigenomics and nutrigenetics is the perspective of the subject. In nutrigenomics, researchers focus on how nutrition intake influences bio-chemical reaction directly related to genes. In contrast, nutrigenetics aims to understand how gene code influences the intake of nutrition and its relevant effects.

Whether one is based in a developed country or a developing country, both nutrigenomics and nutrigenetics are believed to be the solution to infamous and increasingly severe epidemic-chronic diseases. Non-communicable diseases, especially cardiovascular diseases, cancers, chronic respiratory diseases and diabetes, are predicted to be the major cause of death – up to 80% of people will be at risk of survival because of these diseases in developing countries by 2020 [47, 78]. Nutrigenomics and nutrigenetics also cover “inborn errors of metabolism”, which means that one must avoid certain foods containing specific molecules, such as amino acid phenylalanine for phenylketonuria or milk products for lactose intolerance [93].

In order to demonstrate the feasibility of the application of correlating DNA patterns to nutritional factors, some researchers have conducted several trials. For example, in one trial study, participants were randomly separated into intervention groups that received genotype-

based personalised dietary advice and control groups that received general dietary advice. This trial indicated that a diet based on genetics is more useful and is accepted by most participants aged from 20 to 35 [94]. Another example is a study that investigated the impact on consumer behaviour in dietary or exercise of direct-to-consumer genetic testing [13].

Single nucleotide polymorphism (SNP) is the most common type of genetic variation among people. It is a variation in a single nucleotide at a specific position in a DNA building block (genome) and is known to influence nutrients and consequently gene expression, such as mRNA synthesis (transcriptomics), protein synthesis (proteomics), and metabolite production (metabolomics) [91]. The mapping of various health-related factors has been found in many studies, for example in the field of associating diseases and SNPs and in the field of identifying the interaction between nutrition and folate-dependent enzyme polymorphism (folate nutrigenomics) [93, 45].

The correlation between nutrigenomics and nutrigenetics continues to face a number of challenges, particularly regarding three main aspects. First, clinical validity and utility variation in different research organisations leads to opposing arguments [94, 40, 20]. Second, different methods for the risk estimation of diseases and diets, are continuously changing due to breakthrough in genetic research [58, 84]. Third, environmental factors including smoking, diet and activity can affect the measurement of risks or the prediction of diseases. With these challenges, misleading genetic information may cause unnecessary anxiety , hence most services tend to neglect environmental factors or different methods of risk estimation [80]. On the other hand lifestyle recommendations for nutrigenomics and nutrigenetics services tend to be similar, regardless of differences in genotypes.

Scientists have experimented ith human research subjects, correlating candidate genes and nutritional aspects, the indication of which is usually deemed either positive or negative based on the evidence collected [77]. Candidate genes are usually selected through the study of bio-informatics, in which statistical analysis is applied to find risky genes. Aside from genomes, nutrigenetic analysis also considers proteomes, metabolomes and transcriptomes [42].

In this work, the correlation of genetics and nutrition is manually selected and applied based on evidence from publications selected by experts from the company DNAnudge Ltd. With the most influential correlation of genes and nutrients, I have built a new application for a nudge concept called DNAnudge, to be introduced in section 3.2.

Chapter 3

DNAnudge Expert System

In this chapter, the background works prior to my technical invention are introduced. Before working on my models, I first look into supermarket data sources and existing techniques by crawling large datasets from web pages. Following data collection, I design the top level of my system framework with the mindset of a concept called DNAnudge to provide a service. The proposed system is built with consideration of generalisation and scalability. The procedure of obtaining the supermarket data can be divided into several steps: data collection, data processing, generalising the representation of food names from text to numeric form and training a machine-learning classifier to perform categorisation. I explain the data collection and extraction process in Section 3.1. I cover the concept of DNAnudge and present the system framework in Section 3.2. During the process of designing the system, I consider scalability and generalisation, which are explained in Section 3.3. After the data collection, the texts describing the supermarket products are converted into numeric form to create a data set that is ready to train a machine-learning classifier and to generate decision recommendations. This process involves two of my invented models, explained in chapter 4.

3.1 Data Collection and Processing

Data collection aims to access all available data on the Internet. This is challenging due to the fact that structured data on food products are kept within providers' databases. Most information is available on the World Wide Web in an encoded and unstructured format and can be easily accessed. In order to analyse and make use of such information, downloading and extracting are essential steps.

Internet information extraction is an important programming technique in computer science that seeks to extract data records from web pages and to identify items written in Hypertext

Transfer Protocol (HTML), Javascript or PHP [27]. This technique can facilitate data analysis and data integration and has been applied in different research areas such as semantic web [65].

One of the key components in web information extraction is called the wrapper. A wrapper is a programme for data extraction that consists of three approaches [2]: 1) *the manual approach* requires a programmer, supervising the programme to find data extraction patterns by observing the web page and the source code; 2) *wrapper induction* is a semi-automatic approach and requires manual labelling, training templates or extraction rules to apply machine learning techniques; 3) *and automatic extraction:* is an unsupervised approach that does not involve manual labelling, which can scale up the coverage of sites for data extraction.

With the three types of wrapper, two main kinds of data can be extracted [2, 123]: 1) a group of data that is described in a similar format, HTML tags and rendered in a contiguous region; and 2) a list of data presented in sub-trees and under the same parent node with similar repetition of HTML tag structures. Numerous frameworks designed to capture these two types of data has been published in the research literature. For example, information extraction based on pattern discovery(IEPAD) discovers repetitive patterns by matching HTML tag strings and creating a tree structure of the extracted data [21]. In order to reduce computation time and to improve the accuracy of the extracted data, Data Extraction based on Partial Tree Alignment (DEPTA) was proposed to consider not only the source code but also visual information with an alignment technique [21, 138].

Another good example of automatic extraction was proposed in 2015 and is displayed in Figure 3.1. This system architecture contains four main steps [27]:

1. Transforms a web page into a Document Object Model (DOM) tree structure, a cross-platform and language-independent convention for presenting HTML.

2. Data Path Matching (DPM) is used to identify the similarity of the structure of the sub-trees within the DOM structure. This method takes the place of conventional complex tree distance measuring algorithms to ascertain the similarity of the codes of the “data path”.

3. Once the similar sub-trees have been found, the unimportant data is deleted and the overlapping is observed followed by the merging of the section.

4. The Path-Code-Sting Alignment technique has been introduced to align corresponding data items, to transform records into HTML tag strings and to measure the minimum distance between characters.

In this work, a fully automatic data extraction method is not the aim, as I seek instead to extract with a greater accuracy using a specific type of complex online shopping websites

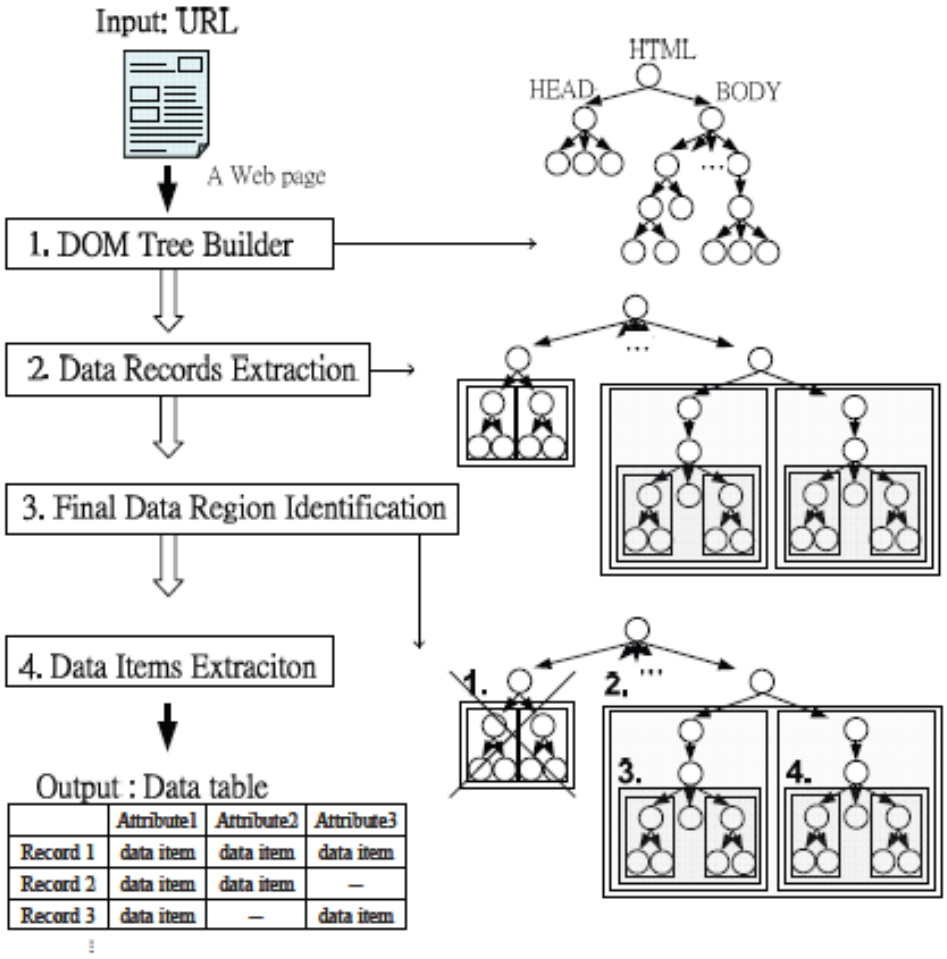


Fig. 3.1 Web information extraction system architecture designed by a Taiwanese team, in which the four main steps of processing are involved. The right side of the figure shows the changes in the data structure along with the processing. [27]

such as supermarket websites. Therefore, I have designed a semi-automatic data extraction technique with some manual extraction rules and labelling.

After developing the wrap web page techniques, extracted data is ready to be applied into the data processing chain. Figure 3.2 shows the schematic of an example of a supermarket database server. The database consists of a large number of data tables containing information about different products. Whenever a user starts a web browser, it retrieves the source code of a web page from the web server by making request using Hypertext Transfer Protocol (HTTP). Within the web browser, it converts the source code (HTML layer) into a presentable web page, which links all the information, images and arranges the layout and functions.

Fig. 3.2 An example of a grocery database server. It consists of three layers: a web-page layer, an HTML layer and a back-end database.

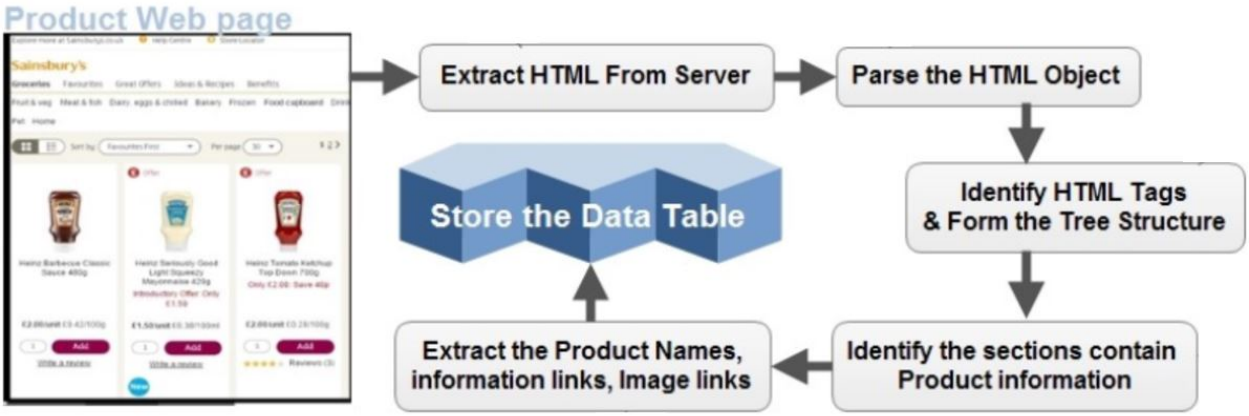


Fig. 3.3 Information extraction steps that are applied to supermarket websites. HTML codes are extracted from supermarket servers, parsed and converted into a tree-structure. The product information is identified from the tree-structured data and stored in a data table in the form of URLs.

Figure 3.3 summarise the data collection and processing of this project into five main steps. First, the HTML document is received from the product web page server. Second, the document is syntactically analysed (or parsed) and converted into meaningful text strings. This step is achieved based on HTML tags or the end sign of the natural language such as a full stop or period. In identifying the HTML tags and their function within all strings of the document, the third step converts the HTML object into a tree model, while each sub-tree contains the texts

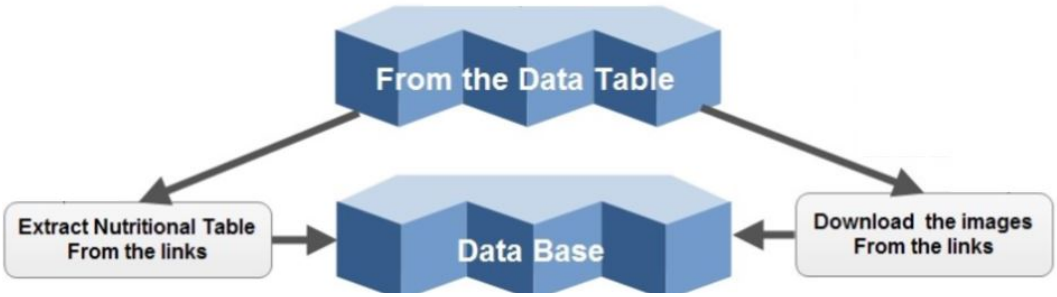


Fig. 3.4 After the key components of supermarket data is identified and stored in the data table, the images are downloaded and the nutritional tables are extracted from the stored URLs.

within the start and end tags of the HTML. In the fourth step, the texts within each sub-tree are compared in terms of their sentence structures or similarities of words. At the same time, the key entity such as the name of a food company or a product is recorded. Similar sub-trees are grouped into the same group and the group with the largest number of key entities is identified as the important section. Finally, all the texts of the important sub-trees are extracted from the model and the desirable information such as image links and names of products are placed into a data table.

Given that different products within a supermarket's web site is presented in the same format and a template of extraction rules for the products is manually designed based on the format, the extraction procedure is executed smoothly and with high accuracy.

Once the data table with the Uniform Resource Identifier (URL) of the product details and images has been generated, the nutritional table can be extracted and the images of each product can be downloaded from the providers. This process is shown in Figure 3.4. All the collected data is locally stored in a Structured Query Language(SQL) database and are assigned an identifier corresponding to the name, image and nutritional table of the product.

Once the names, images and nutritional data has been extracted, data curation takes place, involving identification, transformation and normalisation. Figure 3.5 shows the procedure of the data curation process. The nutritional table consisting of nutritional factors, measurement method and units is treated as the input of this stage. The process of identification has three main objectives. The first is to identify the nutritional factor, which can later be treated as the target of transformation. The second is to identify the measurement method, which helps to understand the format of transformation for later execution. The third is to identify the units, which assists in preparing the step of normalisation. The nutritional tables of different products are transformed based on the nutritional factors and methods of measurement into a collection of tables. Furthermore, the units are normalised to facilitate numerical analysis later.

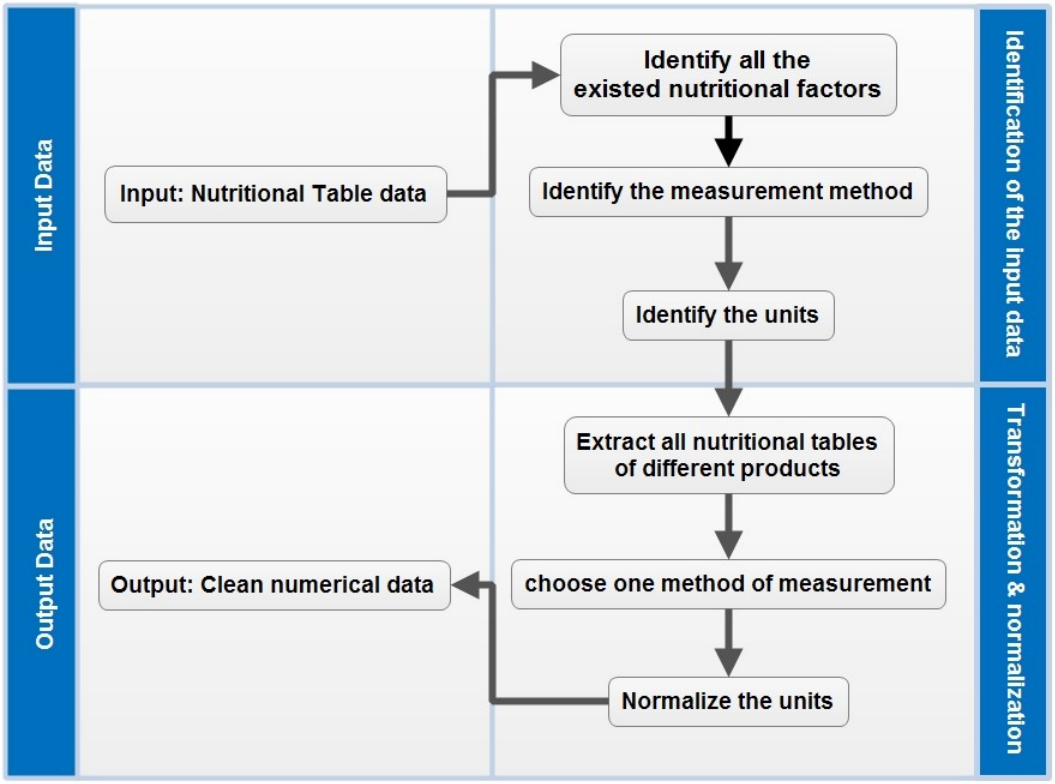


Fig. 3.5 The data is cleaned and normalised before they can be used to train or service a system. This step is referred to as data pre-processing and consists of identifying measurement factors and units and converting all the data into one common format.

3.2 System Framework of DNAnudge for Nutrition Optimisation

In this work, I design a system that combines existing algorithms to construct a practical application for grocery product recommendations based on genetics. This system has been built using two of my invented models for categorisation and recommendation introduced in sections 4.2 and 4.4 and published in IEEE Transactions on Biomedical Circuits and Systems in an article titled "PERSON –Personalised Expert Recommendation System for Optimised Nutrition". [22] I have also extended and written this work in a chapter of the book Trends in Personalised Nutrition, edited by Charis Galanakis and published by Elsevier. [44] In the present section, I offer a brief explanation of this expert system.

DNAnudge is a new concept defined by Professor Christofer Toumazou that seeks to change people's lives from the inside out. Through the study of genetics, it is based on an understanding of the biochemistry within individuals and altering their outer behaviour as a result. Genetics refers to knowledge of genes and genetic variation and froms the basis of how

Fig. 3.6 General framework of DNAnudge, which consists of sourcing DNA knowledge, performing personalised recommendations, designing a service with the mindset of Nudge theory, building an expert system to perform servicing and wrapping the expert system into an application.

all living organisms behave [48]. Nudge theory aims to optimise people's behaviour through small changes in their daily lives. It is clear that there is a perfect match between the two concepts, which in general is to nudge people's behaviour based on what is good for the person according to his or her genes.

Traditional approaches tend to apply nudge theory through policy making or creating a standard operating procedure by governments or companies. However, most approaches are general and not personalised. Furthermore , they conceal the reasoning behind nudge from the target population and ignore the educational opportunity the theory presents.

In this research, I aim to achieve a personalised nudge through expert systems based on individuals' DNA. Moreover, the designed system can provide reasoning through a recommendation system.

The general framework of DNAnudge is shown in Figure 3.6. This framework consists of five main components: DNA knowledge, personalised recommendation, nudge theory, expert system and applications.

This investigation starts by reviewing the knowledge of a section of DNA, including professional publications about genetics. With this DNA knowledge, the research proceeds by discussing how individuals with different genes can benefit from making different decisions and what kinds of recommendations can improve their daily lives through healthier product decisions. Once a good understanding is developed of recommendations for people with various types of DNA, the discussion shifts to how such recommendations can be met in an environment of nudge. A practical way to achieve this nudge environment is to design an expert system and to package this system into an application such as a mobile apps or website.

The food industry is a collective and diverse business that supplies food to consumers. Companies within this industry have become more and more interested in the area of health and life style, particularly in terms of personalised food. With the increase in research and

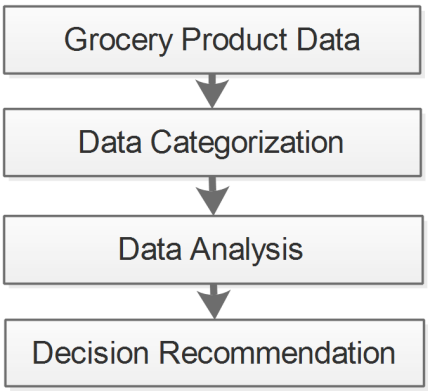


Fig. 3.7 In my expert system, the grocery data is categorised and analysed before being used for decision recommendation.

publications, the fields of nutri-genetics and nutri-genomics are more robust today than ever before, and can now be incorporated with in real industrial applications. However, due to the complexity of available data and information, there remains a gap between industry and academia.

With the aim of filling this gap and creating the first framework for DNAnudge, I propose a Personalised Expert Recommendation System for Optimised Nutrition (PERSON). In the following chapters of this thesis, I will cover in detail how I have designed my proposed systems and applications.

The overview of the system can be separated into four sections, shown in Figure 3.7, First, the grocery product data is collected from websites and second the applied algorithms are used for data categorisation, data analysis and decision recommendation. In the same section, DNN models and other machine learning models are compared and applied to deal with data categorisation due to their outstanding performances with complex data that contains noise. Third, a novel model is proposed to cover data analysis, while a decision recommendation is optimised with a genetic algorithm (GA).

A proposed expert recommendation system framework consists of four main components:

1. A Word Embedding and Padding model is introduced to convert text data into generalised vectors in order to negotiate data variety and unknown grocery product names and to mitigate the out-of-vocabulary problem.

2. A DNN model for product categorisation is covered to cope with the complex features and sequence logic of the grocery product vectors. The selected model can perform categorisation tasks with good accuracy.

3. A decision recommendation model is designed to store the categorisation results, analyse nutritional data and optimise the suggestion provided to the consumer with the designed fitness score and GA.

4. An operational state machine is included that seeks to control the state and to retrain the models of each components during the updating of training data.

The contribution of this work includes, the novel architecture of the personalised expert recommendation system for optimised nutrition (PERSON) based on genetic phenotypes, the utilisation of a deep neural network for grocery product categorisation, and the implementation of a recommendation system using GA.

The method of data collection and processing has been described in this chapter along with a general explanation of the proposed expert system framework. The following chapters focus on detailed processes of the expert system.

Chapter 4.2 describes the scalable food categorisation process using collected data. An overview of the expert system's application and a state machine with an interface for human input is presented in chapter 4.4.

3.3 Generalisation and Scalability

Before I dive into the design and implementation of my proposed model and system, I first discuss the topics of generalisation and scalability so that my applied applications can scale with unknown data that is not in my database.

Generalisation is a very important concept to consider when designing any system. It is a means of extracting general concepts from a targeted instance. The common properties of different elements within domains that consist of shared characteristics are compared. Apart from enhancing the common sections of different elements, it is also important to reduce noises. To verify generalisation, situations are usually created with an examination scoring. While this concept is broad, setting up a scope is an important first step. In this project, I focus on the generalisation of the feature expression of the text of supermarket food names. This significantly influences the machine learning model and system functionality for food categorisation.

In addition to generalisation, scalability is also significantly affected by the data feature presentation. The scalability of a system is defined as the capability to handle and accommodate a growing amount of work [14]. A system is scalable if it can be applied in large situations, that is, it is able to take a substantial input data sets and produces a high number of outputs. For example, a search engine not only has to scale with the number of users but also the number

of search indexes. Among the various ways that exist to measure scalability, my proposed system focuses on the area of generation scalability. This describes the ability of a system to use components from different vendors or sources [39].

In this work, vertical and horizontal scaling are considered with in the perspective of software and algorithms, in contrast to the hardware perspective that considers the number of computing nodes or memory in each nodes. For horizontal scaling, I consider the system's ability to cope with different food names, including English words that have never appeared in the training sets. As for vertical scaling, I consider the amount of data that a system can take each time in order to provide services.

After vertical scaling, my designed architecture model is capable of retraining and reproduction. Therefore it can be very easily duplicated onto different servers or cloud platforms. Once the system is deployed into the cloud platform, it takes advantage of the scalability of the computing power of the platform.

For horizontal scaling, a word-embedding technique and a single-layered neural network are designed to allocate each English word into a multidimensional space vector. However, with the limited number of English words within a training data set, the issue of facing unrecognisable new English word frequently appears. In seeking to increase vocabulary size and cover new common words, I thus also apply unsupervised learning on word-embedding by incorporating with Google News data instead of food name data. This is further discussed in detail in the next section.

Chapter 4

Personalised Expert Recommendation System for Optimised Nutrition

In this chapter, I explain the process of converting text data into a numerical representation. The numerical expressed datasets are then treated as inputs to a machine learning classifier. I also describe the techniques and information required for the representation procedure, DNN classifier, genetic algorithm and the design of my recommendation system in detail. Section 4.1 explains the word-embedding method that is applied to convert text data into numeric data. Treating word-embedded products' names as inputs, the extension of the neural network architecture and other machine-learning models are described and compared in Section 4.2. I describe how the genetic data is expressed and introduce an algorithm to optimise my recommendations, called a genetic algorithm, in Section 4.3. With the genetic data and categorised grocery food data as inputs, I explain the design of my recommendation system in Section 4.4. I lastly put together all the models introduced in this chapter in an application framework in Section 4.5.

4.1 Background of Scalable Food Categorisation

Word embedding is a distributed representation of words in numerical vectors and it has demonstrated outstanding performance in word similarity tasks and word analogy/analysis in deep learning studies [86]. The general method of the bag-of-word (BoW) model has proved unable to classify short texts [122], as it ignores the order and semantic relations between words. In contrast, more advanced models such as neural network language word embedding have been shown to preserve semantic types and syntactic relationships. Furthermore, embedding words into vectors with real numbers from one dimension per word to a continuous vector space

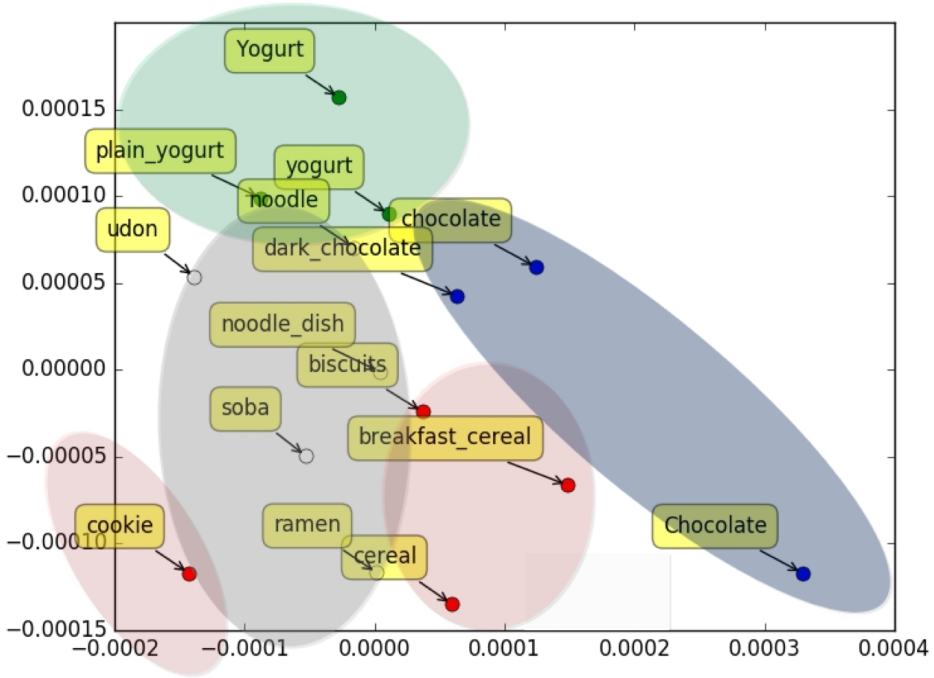


Fig. 4.1 Product names are converted into 300-dimensional numerical vectors using a word embedding technique. For the purpose of demonstration, the t-SNE technique was applied to project a 300-dimensional vector into a two-dimensional vector. In this plot, the vertical and horizontal axes are combined numerical representations of multiple dimensional data from the original 300-dimensional vector. The circles with different colours represent the different meaning groups of the covered texts.

ensures the low dimension of vectors [85]. The vectors of words in the embedding space have similar semantic relationships [86], enabling the advancement of clustering tasks in Natural Language Processing (NLP). In this work, the continuous Skip-gram model is used to generate word embedding, which consists of an input, a projection and an output layer. The input layer observes each current word and predicts a certain range of words located before and after. The increased range improves the quality of word embedding, but causes a trade-off at the expense of computation load.[51]

The Skip-gram word embedding model is trained with 100 billion words from Google News. The word vector has 300 dimensions and covers three million unique words [85]. With dimension reduction techniques such as t-distributed stochastic neighbour embedding (t-SNE), the word vector can be projected down to two dimensions for the purpose of visualisation [76], as shown in Figure 4.1. The figure shows the visualisation of words from the case study data

of this work. Name entities with similar semantic types can potentially be clustered into one category and create different regions for classifications.

With word embedding techniques applied in my system, I am able to avoid the out-of-vocabulary problem. This issue often occurs when the database dictionary does not recognise a word pattern and can not convert the word into numerical expression. In the training process of embedding dictionary words, I incorporate not only the available food product text data but also whole Google News data sets. This extends my dictionary to not only cover the words of current food products but also the future known texts of food product names. This is essential for my system to generalise my inputs and scale with future data.

The following paragraphs of this section explain the categorisation process using the generalised word-embedding data sets once the food data has been extracted from the Internet and converted into numerical vectors. Different machine learning models are discussed to implement the model for food product categorisation, including a Linear Classifier, a Support Vector Machine (SVM), a Convolutional Neural Network(CNN) and a modified Recurrent Neural Network (RNN) called Long Short-Term Memory(LSTM).

Categorisation aims to solve objects classification, recognition, differentiation and understanding [30]. During categorisation, objects are grouped into categories based on certain purposes, such as the relationships between ideas or the meanings of the categories and objects. Categorisation plays an important role in computer science and can be applied in many fields, such as natural language prediction, inference and decision processes. [43].

Machine learning is a key field in computer science that provides solutions to categorisation tasks. It is essential to build a supervised model to solve the categorisation problem. The general procedure of building a supervised model begins with separating available data into training sets and testing sets. Then a strategy is required to find the architecture of the model that performs classification [126]. The model is trained through observing the training data set, making decisions and receiving punishment if the decision is incorrect. In each observation, the data sets can be treated as a set of quantifiable properties, which technically are referred tos as variables or features. Considering my project as an example, the food products' names are converted into vectors in space through word embedding as previously shwon in Figure 4.1. The vectors contain multiple numbers that represent the values of the axes in different dimensions. The space location that stores these numbers corresponds to the variables or features.

To build the model architecture, a strategy has to be defined through specifying an optimisation procedure and a mathematical form of the problem. Once the model has been found, it is treated as a classifier that compares observations by means of similarity. The strategy here is called the algorithm in mathematics domain which is thought to constitute an implemented mathematical function that maps input data to a category [32].

The ability of a classifier to adapt to unknown future data represents the most important factor once the model has been built and trained. This is done through examining the model using a testing data set. The property and result are highly contingent on the characteristics and the representation of the data. The procedure by which to measure this property is called evaluation. Measuring precision and recall are popular approaches, while simply performing an accuracy test is the most common way to evaluate a model [109].

This experiment started with a relatively simple classifier called a 'linear predictor function'. This type of classifier is frequently applied and can be adapted to a large number of algorithms to categorise a problem. The function can be expressed as shown in Equation 4.1 where $\mathbf{X}_i$ is the value from instance **i** of the input feature vector and $\mathbf{W}_k$ is the weight of each **k** output category vector.

$$\textbf{score}(\mathbf{X}_i, \mathbf{k}) = \mathbf{W}_k \cdot \mathbf{X}_i \tag{4.1}$$

The $\textbf{score}(\mathbf{X}_i, \mathbf{k})$ is the score for assigning the input instance **i** to category **k**. The prediction is made by selecting the one with the highest score. The algorithm with this basic set-up is referred to as a linear classifiers. Such classifiers are known to work well with high-dimensional vectors, and require less time to train. These properties are well suited in my case due to the fact that my input product name vectors have 300 dimensions, which is a high-dimensional vector problem [136]. However, the projected location of different food names in groups from Figure 4.1 shows that a linear classifier has limited performance due to the shape of the clusters and to non-linearity within the dimensions of the data.

To improve the classification's performance, I implemented a different classifier called a support vector machine (SVM). SVM utilises a kernel function that can detect non-linearity within dimensions of data. The SVM model represents data as points in space and seeks to separate data points through creating gaps of categories. When facing a new data point, it predicts the point category based on the side of the gap that the point location lands. When dealing with non-linear classification, SVM can apply the kernel function. This maps its input data into another dimensional feature's space, and allows the algorithm to fit the model there. Note that the transformed feature space is in a higher dimension, which increases the generalisation error of the model. To build a SVM model, the first step to establish hyperplanes. The definition of any hyperplane is defined as a set of input $\mathbf{X}_i$ following Equation 4.2 [33].

$$\mathbf{W}_k \cdot \mathbf{X}_i - \mathbf{b} = 0 \tag{4.2}$$

where the weight $\mathbf{W}_k$ is now treated as a normal vector to the hyper plane, while $\mathbf{X}_i$ is the input data and the bias **b** refers to the offset of the hyper plane. This hyper plane provides an

important basis for a SVM to create a boundary that performs a classification task [83]. A SVM normally employs two approaches when classifying data: a hard margin approach and a soft margin approach. The suitable approach is contingent on whether the data is linearly separable or non-linearly separable. For the case of linearly separable, the hard margin approach is applied, where parallel hyperplanes are set. The parameter such as weight $\mathbf{W}_k$ and bias $\mathbf{b}$ is tuned with the aim of maximising the distance of the hyperplanes while keeping the data on the correct side. For example, Equation 4.2 can be used to express a two-category-SVM as shown in Equations 4.3 and Equation 4.4 [50].

$$\mathbf{W}_k \cdot \mathbf{X}_i - \mathbf{b} = 1 \tag{4.3}$$

$$\mathbf{W}_k \cdot \mathbf{X}_i - \mathbf{b} = -1 \tag{4.4}$$

For a two-category-SVM, Equation 4.3 is used to classify any data point for label 1 and Equation 4.4 is used to classify any data point for label -1. For any tuning process of weight $\mathbf{W}_k$ and bias $\mathbf{b}$, it aims to keep all data points with label $\mathbf{y}_i = 1$ above the boundary defined by the hyperplane of Equation 4.3 and all data points with label $\mathbf{y}_i = -1$ below the boundary defined by the hyperplane of Equation 4.4. This process is represented by Equations 4.5 and 4.6.

$$\mathbf{W}_k \cdot \mathbf{X}_i - \mathbf{b} \geq 1 \quad for\, \mathbf{y}_i = 1 \tag{4.5}$$

$$\mathbf{W}_k \cdot \mathbf{X}_i - \mathbf{b} < -1 \quad for\, \mathbf{y}_i = -1 \tag{4.6}$$

Another objective of my tuning process is to maximise the distance of the two hyperplanes which can be expressed as the value of $\frac{2}{||W_k||}$. This also means the optimisation process seeks to minimise $||W_k||$ while keeping Equations 4.5 and 4.6 true. Note that during the process of optimisation the max-margin hyperplane is tuned by input X_i, the inputs of which are termed the support vectors.

For the case of non-linearly separable data, the soft-margin approach is applied in the SVM. A common way of applying soft-margin SVM is to introduce a hinge loss function as shown in Equation 4.7.

$$\mathbf{max(0,1\text{-}y}_i\mathbf{(W}_k \cdot \mathbf{X}_i - \mathbf{b)}\ \mathbf{)} \tag{4.7}$$

In Equation 4.7, y_i is the target data label that takes note of feature input X_i data. In the establishment of a two-category-SVM, Equation 4.7 is zero when the data X_i are located on the correct sides of the margin and outputs a value when the data X_i are located on the incorrect

sides. The output value for the latter scenario is proportional to the distance from the defined margin.

With the introduction of hinge loss function in Equation 4.7, the objective function can be set up to aim to minimise Equation 4.8.

$$\frac{1}{n}\sum_{i=1}^{n}\textbf{max(0,1-}\mathbf{y}_i(\mathbf{W}_k\cdot\mathbf{X}_i-\mathbf{b})\textbf{)}+\lambda||\mathbf{W_k}||^2 \tag{4.8}$$

where n is the total number of input data, and the parameter λ is used to determine the consideration of margin-size. When the parameter λ is small, the soft-margin SVM behaves similarly to hard-margin SVM. In order to improve the accuracy of non-linear classification, the kernel function is introduced [4]. The kernel function seeks to project data points into another dimension with a projection kernel function $\phi(X_j)$ as shown in Equation 4.9.

$$\mathbf{k}(\mathbf{X}_i,\mathbf{X}_j)=\phi(\mathbf{X}_i)\cdot\phi(\mathbf{X}_j) \tag{4.9}$$

Moreover, the weight vector is transformed into Equation 4.10.

$$\mathbf{W}_k=\sum_{i=1}^{n}\mathbf{c}_i\mathbf{y}_i\phi(\mathbf{X}_i) \tag{4.10}$$

With Equations 4.9 and 4.10 function $(f(\mathbf{c}_i)\ \ for\ \ i=1,2,\ldots\ to\ n)$ is maximised and expressed in Equation 4.11.

$$f(\mathbf{c}_1,\mathbf{c}_2,\ldots,\mathbf{c}_n)=\sum_{i=1}^{n}\mathbf{c}_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{y}_i\mathbf{c}_i(\phi(\mathbf{X}_i)\cdot\phi(\mathbf{X}_j))\mathbf{y}_j\mathbf{c}_j \tag{4.11}$$

This is under the constraint of $\sum_{i=1}^{n}\mathbf{c}_i\mathbf{y}_i=0$ and $0\leq\mathbf{c}_i\leq\frac{1}{2n\lambda}$. The coefficients $\mathbf{c}_i$ are obtained from the above objective functions and b is solved with Equation 4.12.

$$\mathbf{b}=\mathbf{W}_b\cdot\phi\mathbf{X}_i-\mathbf{y}_i=\sum_{k=1}^{n}\mathbf{y}_k\mathbf{c}_k(\phi(\mathbf{X}_k)\cdot\phi(\mathbf{X}_i))-\mathbf{y}_i \tag{4.12}$$

With the collected coefficients $\mathbf{c}_i$ and $\mathbf{b}$, the model can be expressed to classify new points with function $(f(\mathbf{c}_i)\ \ for\ \ i=1,2,\ldots\ to\ n)$ and Equation 4.13.

$$Classification=sgn(\mathbf{W}_k\cdot\phi(\mathbf{X}_i)-\mathbf{b}) \tag{4.13}$$

SVM works well with relatively small data sets, due to the fact that it considers the whole picture and generates a globally optimal solution. However, when dealing with complex large data sets, SVM presents limited performance compared to Deep Neural Networks(DNNs),

particularly in the study of Natural Language such as text data processing [124, 61]. In the experiment of this work, the extracted food data sets from the web were relatively large and contained texts such as food names. Therefore, I extend my research to apply DNNs models.

DNNs are extensions of artificial neural networks, which are computing models inspired by biological neural networks in animal brains [96]. Artificial neurons are connected and free to transmit signals from one to another, while each neuron processes the received signal and sends the results to the output.

Artificial neural networks are formed by structures called artificial neurons. The artificial neuron has an internal state that is mathematically called an activation function and produces outputs based on the activation function, input and another element called weights [9]. The weights represent the key components in performing a process called learning, which tunes the values of weights according to the detection of an error between the produced output and the target output [137, 5]. Equation 4.14 represents a single neuron with label k, input X_i, a weight vector W_k and a bias vector b_k.

$$\mathbf{net}_k = \mathbf{W}_k \cdot (\mathbf{X}_i) - \mathbf{b}_k \tag{4.14}$$

This equation aims to mimic the behaviour of the neuron receiving signal $\mathbf{X}_i$ and summarise its current status value $\mathbf{net}_k$. The status value is sent through an activation function f and output signal $\mathbf{y}_{i,k}$ as shown in Equation 4.15.

$$\mathbf{y}_k = \mathbf{f}(\mathbf{net}_k) \tag{4.15}$$

In general learning rule, the action of learning is through modifying the weight vector $\mathbf{W}_k$ shown in Equation 4.16.

$$\mathbf{W}_k^{t+1} = \mathbf{W}_k^t + \Delta\mathbf{W}_k^t \tag{4.16}$$

where t is the current time step and $t+1$ is the next time step. The weight of the next time step $\mathbf{W}_k^{t+1}$ is obtained through the sum of the weight of the current time step $\mathbf{W}_k^t$ and the modification vector $\Delta\mathbf{W}_k^t$. Equation 4.17 represents the process of vector modification of the learning signal $\mathbf{e}_k^t$, which is a function of current weight $\mathbf{W}_k^t$, current input $\mathbf{X}_k^t$ and target vector $\mathbf{T}_k^t$.

$$\mathbf{e}_k^t = f(\mathbf{W}_k^t, \mathbf{X}_k^t, \mathbf{T}_k^t) \tag{4.17}$$

In other words, this learning signal is measured based on how far away from the output signal $\mathbf{y}_k^t$ to target vector $\mathbf{T}_k^t$, given $\mathbf{W}_k^t$ and $\mathbf{X}_k^t$.

Finally the $\Delta \mathbf{W}_k^t$ is obtained from Equation 4.18, where η is a defined learning rate.

$$\Delta \mathbf{W}_k^t = \eta \mathbf{e}_k^t \mathbf{X}_k^t \tag{4.18}$$

When the learning signal is consistently low and the training is completed, Equations 4.14 and 4.15 are used to perform the classification shown in Equation 4.19.

$$Classification = \mathbf{y}_k = f(\mathbf{W}_k \cdot \mathbf{X}_i + \mathbf{b}_k) \tag{4.19}$$

DNNs are artificial neural networks with multiple layers, permitting the expression of complex non-linear relationships to be modelled [4]. The more complicated the features of the input data, the more layers of DNN need to be applied in order to improve accuracy for classification. Furthermore, a variation of DNNs architectures provides a good alternative to simply scaling the layers. DNNs have two common architectures: Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNNs have many variations and are part of feedforward neural networks that are frequently used for computer vision and natural language processing tasks [31, 63, 66]. The basic form of CNN consists of one or more convolutional layers, pooling operations and fully connected layers [70]. To understand CNNs' operation, I first use a single-layer CNN model and scale the number of layers for the experiments. A single-layer CNN consists of a convolutional layer, a kernel function, a max pooling operation and a fully connected layer,as shown in Figure 4.2.

With the word-embedding method introduced in the previous section, food product names are converted into numerical vectors $\mathbf{x}_{i,j}$, where i represents the identification of the data I am looking at and j refers to word identification. For example, for the word vector of "Milk" in the product name "Milk Giant Chocolate" with a product id of 5, the notation is presented as $\mathbf{x}_{2,5}$. Note that to present the full name of a product I can simplify the notation with Equation 4.20, where p is the number of words of a product.

$$\mathbf{X}_i = x_{i,j} \quad for\ j = 1, 2..., p \tag{4.20}$$

The kernel window takes a pre-defined number of inputs, multiplied by a set of weights $\mathbf{W}^{kernel}$ plus a kernel bias $\mathbf{b}_c$ and shifted to a new location. After being shifted to a new location, the window will take another set of inputs but be multiplied with the same weight $\mathbf{W}^{kernel}$. For example, setting up the pre-defined number of input words as 5, kernel window as 2 and shift location length as 1, the kernel function can then be expressed as Equation 4.21, where $T = p - 1$.

$$\mathbf{u}^t = \mathbf{W}^{kernel} \mathbf{X}^t_{window} + \mathbf{b}_c \quad where\ t = 1, 2, ...T \tag{4.21}$$

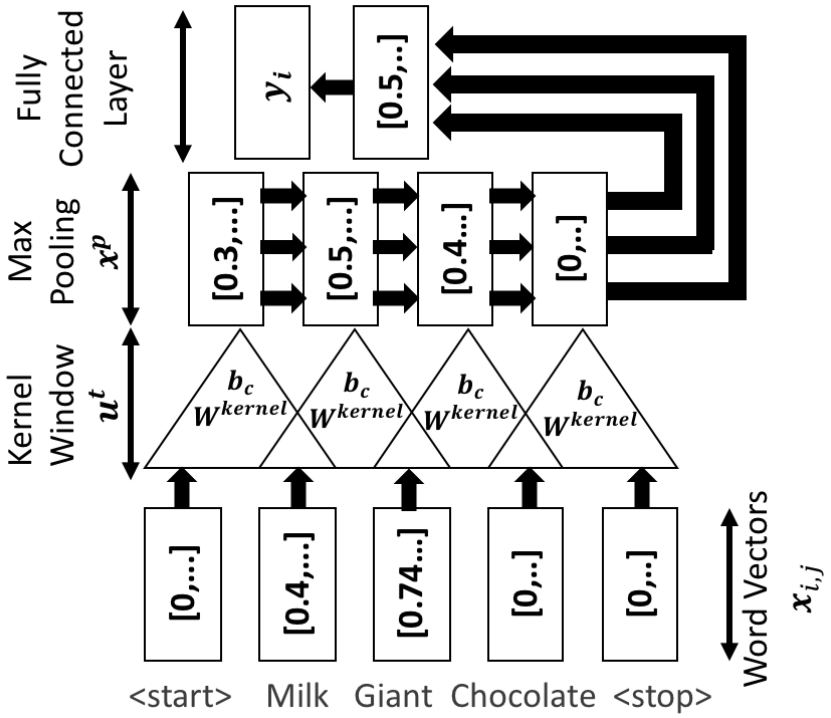


Fig. 4.2 The architecture of the CNN model, which consists of a kernel window, a max-pooling and a fully connected layer. The model takes inputs from word vectors $\mathbf{x}_{i,j}$ and output label vector $\mathbf{y}_i$.

Specifically for the case of $t = 1$, we have Equation 4.22.

$$\mathbf{u}^1 = \mathbf{W}^{kernel}\mathbf{X}^1_{window} + \mathbf{b}_c = \mathbf{W}^{kernel}_1\mathbf{X}_{<start>} + \mathbf{W}^{kernel}_2\mathbf{X}_{Milk} + \mathbf{b}_c \tag{4.22}$$

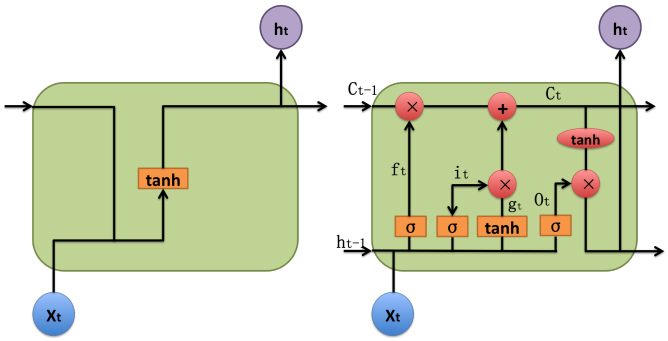
Specifically for the case of t =2, we have Equation 4.23.

$$\mathbf{u}^2 = \mathbf{W}^{kernel}\mathbf{X}^2_{window} + \mathbf{b}_c = \mathbf{W}^{kernel}_1\mathbf{X}_{Milk} + \mathbf{W}^{kernel}_2\mathbf{X}_{Giant} + \mathbf{b}_c \tag{4.23}$$

The size of $\mathbf{W}^{kernel}$ can be scaled based on the defined size of the windows. The max pooling operation takes $\mathbf{u}^t$ as its input and executes element-wise max value selection, as shown in Equation 4.24.

$$\mathbf{x}^p = [\max_{1\le t\le T} u^t_1\ ,\ \max_{1\le t\le T} u^t_2\ ,\ \dots\ ,\ \max_{1\le t\le T} u^t_p]^{Transpose} \tag{4.24}$$

This operation aims to take the maximum value of each element's location of all the output vectors of the previous stage. As an example, looking at the first element of each $\mathbf{u}^t$ shown in Figure 4.2, the value can be obtained by Equation 4.25.



(a) RNN (b) LSTM

Fig. 4.3 The internal structure of RNN and LSTM. The RNN on the left side of the figure has a much simpler structure; however, it has the issue of information loss due to vanishing gradients. The LSTM model on the right-side solves the problem by introducing a gating mechanism in its structure.

$$\max_{1\leq t\leq T} u_1^t = max[0.3, 0.5, 0.4, 0] = 0.5 \tag{4.25}$$

Finally, the output $\mathbf{y}_i$ can be defined by Equation 4.26.

$$\mathbf{y}_i = f(\mathbf{W}^{FullyConnected}\mathbf{x}_p + b_f) \tag{4.26}$$

With the similar operation of simple artificial neural networks, the fully connected layer takes $\mathbf{x}_p$ as input, multiply with a weight $\mathbf{W}^{FullyConnected}$, add a bias b_f and wrapped with an activation function f. The output $\mathbf{y}_i$ is a vector that is used to perform classification for input $\mathbf{x}_i$ of the $i-th$ product.

CNNs demonstrate excellent performances in extracting the features of different positions within data and can learn relationships between positions and features through an operation called max-pooling [128]. However, they are not designed to handle sequences of data or to capture long-term dependencies. On the other hand, RNNs are specialised for sequential modelling and hence ideal for learning word dependencies [112]. In this work, one type of RNNs is applied due to the significance of dependency between words in product categorisation.

Common traditional RNNs have simple structures such as a single hyperbolic tangent layer as shown in Figure 4.3-a. In the standard RNN model with a long chain of repeating components, vanishing gradients for the parameter updates becomes a serious issue for training, which leads to the introduction of long short-term memory recurrent neural network (LSTM-RNN) [54]. The centre of the LSTM cell is the memory sub-cell and the information can be removed or added to the cell state through the designed gates, as shown in Figure 4.3-b. The operation of a single LSTM cell is described in Equations 4.27 to 4.32.

$$\mathbf{fo}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{4.27}$$

$$\mathbf{in}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{4.28}$$

$$\mathbf{ou}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{4.29}$$

In the above Equations 4.27 to 4.29, $\mathbf{x}_t$ is the input vector of current time step t to the LSTM-RNN cell, while $\mathbf{h}_{t-1}$ is the vector of the previous hidden layer state. These two vectors are put together and weighted by the forget gate weight matrix $\mathbf{W}_f$, input gate weight matrix $\mathbf{W}_i$ and output gate weight matrix $\mathbf{W}_o$. The weighted vectors are summed up with the biases of the same gate type, therefore $\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t]$ is summed with the forget gate bias $\mathbf{b}_f$, $\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t]$ is summed with the input gate bias $\mathbf{b}_i$ and $\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t]$ is summed with the output gate bias vector $\mathbf{b}_o$. The summed results are taken as inputs to sigmoid activation functions σ and form the outputs of the three gates: forget gate output vector $\mathbf{f}_t$, input gate output vector $\mathbf{i}_t$ and output gate output vector $\mathbf{o}_t$. In other words, each gate is formed from a sigmoid layer and a point-wise multiplication operator that can output a value from 0 to 1. The decision of passing the information is decided based on the output value.

$$\mathbf{g}_t = \tanh(\mathbf{W}_g[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_g) \tag{4.30}$$

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \tag{4.31}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \tag{4.32}$$

The actual input information is described in Equation 4.30. Similarly to Equations 4.27 to 4.29, $\mathbf{h}_{t-1}$ and $\mathbf{x}_t$ are put together and weighted by the input weight matrix $\mathbf{W}_g$. Hyperbolic tangent activation function takes the sum of the weighted result and the input bias $\mathbf{b}_g$ and generates input state $\mathbf{g}_t$. The internal current state coefficient $\mathbf{C}_t$ is evaluated through the sum of the forget gate output vector $\mathbf{fo}_t$ multiplied by the previous state $\mathbf{C}_{t-1}$ and the input gate output vector $\mathbf{in}_t$ multiplied by the input state $\mathbf{g}_t$, as shown in Equation 4.31. Finally, the hidden layer output state $\mathbf{h}_t$ is produced by the activation of $\mathbf{C}_t$ multiplied by the output gate vector $\mathbf{ou}_t$ in Equation 4.32. Note that $\mathbf{h}_t$ is treated as the output of the LSTM-RNN cell as well as the recurrent input of the same cell in the next time step.

In fact, the single layer of the LSTM RNN already has deep architectures, as it is treated as a feed-forward neural network with the recurrent connection that loops multiple layers in

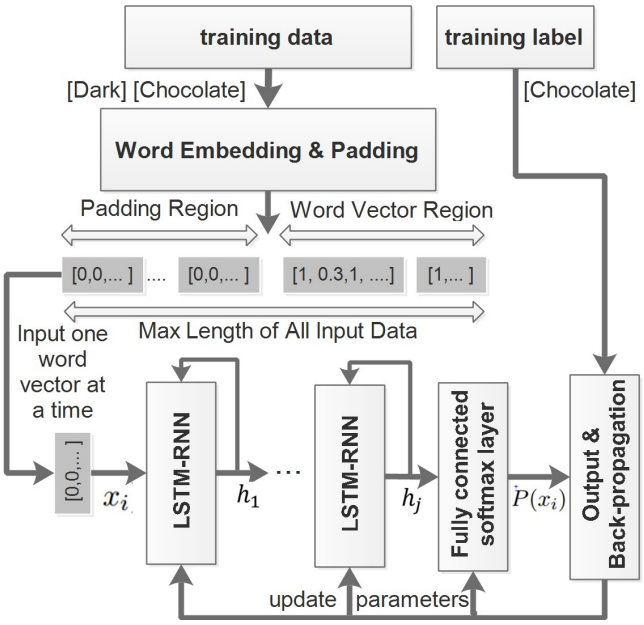


Fig. 4.4 The architecture of the DLSTM model is built with multiple layers of the LSTM model followed by a fully connected softmax process layer. It takes inputs of numeric vectors from training texts that are processed by word embedding and padding. The model is trained by comparing the training label and the output of the model for each training step.

time, with each layer sharing the same parameters. However, this model has limited features due to the fact that the information is processed by a single nonlinear layer through multiple given time instants before contributing the output. Deep LSTM-RNNs are formed using many layers of the LSTM. Deep LSTM-RNNs possess both properties of processing through time and layers, where the parameters are more optimally used via the distribution of the meaning over the space of the multiple layers [113, 55].

4.2 Designed Architecture of Scalable Food Categorisation

In this work, LSTM is applied to capture the semantic dependency of product names for the classification of categorisation. The nature of product names is complex in terms of the varying importance of different sequences. For instance, dark chocolate is a type of chocolate with supporting information dark, while chocolate biscuit is a type of biscuit with supporting information chocolate. The word chocolate represents a very different type of information in different scenarios, indicating that a different sequence causes a word to play a very different role in semantic classification.

Figure 4.4 shows my architecture of the Deep LSTM (DLSTM) model. The input data is converted into word vectors and padded into the maximum length of all training data. At each time instance, one word vector is fed into the input $\mathbf{x}_i$ of the DLSTM, where i is the sequence of time instance. LSTM cells output values to hidden layer $\mathbf{h}_j$ at each layer of the DLSTM, where j is the number of layers of the LSTM RNN. Finally, the last hidden layer $\mathbf{h}_j$ is connected to a soft-max layer expressed as Equation 4.34, where $\mathbf{u}_g$ is the output of the fully connected layer before soft-max function and G is the number of predictions of the classification label.

$$\mathbf{h}_j \odot \mathbf{W}_s = \mathbf{u}_g \tag{4.33}$$

$$Pre(\mathbf{x}_i) = \frac{e^{\mathbf{ug}}}{\sum_{g=1}^{G} e^{\mathbf{ug}}} \tag{4.34}$$

$$\mathbf{loss} = -\frac{1}{n}\sum_{i}^{N}\sum_{k}^{C} Pre_k^d(\mathbf{x}_i)\log(Pre_k(\mathbf{x}_i)) \tag{4.35}$$

The LSTM model is trained with cross-entropy over the training data. In Equation 4.35, **loss** represents the loss vector, $Pre_k(\mathbf{x}_i)$ is used to denote the distribution of the model prediction, and $Pre_k^d(\mathbf{x}_i)$ is the distribution of the inputs. N is used to describe the total number of training data and C is the number of classification labels. The Adam method is used to achieve stochastic optimisation and to update all parameters in each layer of the DLSTM-RNN and the soft-max layer [67].

In order to generalise the experiments and test the system, three databases are incorporated into this work, which are respectively: consumer-contributed, government-organised and corporation-created. For the consumer-contributed database, Open Food Facts, a global packaged food database with contributions from consumers is chosen. By February 2017, Open Food Facts covered 135,891 of products worldwide [41]. For the government-organised database, the United States Department of Agriculture (USDA) database is selected. The USDA database is a public and private partnered organised database, which aims to provide support to the development of consumer applications. By February 2017, USDA database covered 184,022 U.S. products' data [35]. For the corporation-created database, Tesco is chosen. Tesco is a British grocery retailer that has branches in different countries. For research purposes, data from Tesco are gathered from both on-product labels and public Tesco supermarket websites. A total of 23,518 food products of interest were collected in October 2016. The gathered information include product names and nutrition facts tables [49]. After comparing the results of the DLSTM with the three databases, CNN, SVM and linear classifier are also considered to demonstrate the advantages of selecting the structure of the LSTM model.

Table 4.1 The performance of three different DLSTM model datasets with different numbers of layers.

	Open Food Facts	USDA	Tesco
1-layer LSTM	41.8%	40.3%	81.2%
2-layer LSTM	41.9%	45.2%	83.8%
3-layer LSTM	42.1%	46.2%	84.0%

Table 4.2 Testing the accuracy of different machine learning structures on the Tesco training dataset.

	LSTM	CNN	SVM	Linear Classifier
1-layer	81.2%	72.2%	72%	46%
2-layer	83.8%	72.8%	X	X
3-layer	84.0%	72.0%	X	X

All of the product names from the three datasets are used as the input data of the DLSTM-RNN model. In order to compare differences between databases and categories, only some of the common categories are chosen to be included in the experiments. The selected categories are Cereals, Biscuits, Cookies, Yoghurts, Chocolate, Rice, Noodle and Pasta. Due to the complexity of common subgroups of Tesco and Open Food Facts, the datasets are further grouped as: Cereals & Biscuits & Cookies, Yoghurts, Chocolate and Rice & Noodle & Pasta.

After removing some corrupted data, the actual number of products included in the experiments are 942 from Open Food Facts, 9,126 from USDA data and 2,012 from Tesco. Each item of data consists of a category tag, a name and a nutrition facts table. The data is randomly separated into 60% training data and 40% testing data. With random permutations cross-validation, the accuracy is shown in Table 4.1. The results show a good trace of improvement with more layers of the LSTM-RNNs. The Tesco datasets present high accuracy due to the organised order of product names, which sequenced as brand name, flavour and food name. In contrast, some USDA and all Open Food Facts data follows random orders. Given that the Open Food Facts data is manually created, some data does not cover the full name of the product. The cross-validation method is time-consuming for deep learning models, therefore training beyond three layers of the DLSTM is not covered in this work at the current stage. Note that the incorrect classifications of products are manually adjusted through an interface during updates and the updated data is fed to the stage for personalised grocery decisions during the training stage to avoid errors in product recommendations due to mis-classification.

CNNs and LSTM-RNNs are popular architectures in DNNs. Comparison of the two architectures of DNN and other simpler models such as SVM and linear classifier based on the Tesco dataset present the same experimental set-up as the previous experiment. The results are shown in Table 4.2. The experiment's results show that LSTM-RNN outperforms CNNs, SVM and linear classifiers. This is due to the fact that CNNs and the two other models are unable to capture the dependencies of words within product names, whereas LSTM-RNN can. Comparing between the different numbers of layers of CNN does not show traces of improvement, either. Three-layer LSTM is selected for implementation in my proposed system due to its superior performance.

Table 4.3 An example of the rs5400 genotypes sourced from SNPedia. This demonstrates the different strength of sugar consumption-ability between genotypes.

Genotype	Magnitude	Summary
(C; C)	0.1	Normal sugar consumption
(C; T)	1.7	Significantly higher sugar consumption
(T; T)	1.8	Significantly higher sugar consumption

4.3 Background of Genetic Mapping and Recommendation Optimisation

Deoxyribonucleic acid (DNA) carries the key instructions of genetics. Observing variations in DNA between individuals forms the basis to identifying differences and the needs for personalisation in nutritional advice. The most common means of observing such variation is to study single nucleotide polymorphisms (SNPs), which are the different DNA sequences between members of humans as regards the four genome patterns adenine (A), thymine (T), cytosine (C) and guanine (G). SNPs are known to influence how a person absorbs and metabolises nutrients and are frequently used to find evidence to confirm assumptions, thereby forming the basis of nutrigenetic sciences [42, 34].

A phenotype is the observable characteristic of an organism in biology that is caused by the expression of the genetic code, the genotype, and environmental factors [28]. The distinction of genotype-phenotype was first proposed by Wohelm in 1911 [59], and was later extended by many other researches [73, 6]. In general the relationship can be expressed as in Equation 4.36:

$$\mathbf{G} + \mathbf{E} + \mathbf{GE} \approx \mathbf{P} \tag{4.36}$$

where **G** is the genotype, **P** is the phenotype, **E** represents an environmental factor and **GE** represents the interaction between the genotype and the environment. Genotype data can be found in SNPedia with reference to papers sourced from Pubmed, where new research results and publications are continuously updated [19]. Table 4.3 presents an example of the genotypes of rs5400 from SNPedia. The genotype rs5400 is a gene location and three variations with different expressions are associated with sugar consumption.

Environmental factors are defined differently in distinct practical situations, but are usually directly related to the genotype. For example if the genotype focuses on sugar, one can regard individual sugar intake as the environmental factor. In most practical cases, the genetic phenotype is defined as the observable characteristic of the performance of a person's ability. Therefore in this work I consider one's ability to consume five selected nutritional factors:

Table 4.4 Direct mapping of the phenotype to the nutritional threshold of a grocery category. l represents the level of ability of the consumption of a nutrient n. **P** is the phenotype and **T** is the nutritional threshold.

	n = Energy	**n** = Fat	**n** = Salt	**n** = Sugar	**n** = Protein
l = *Good*	$\mathbf{P}_{\mathbf{Good,Energy}} \rightarrow \mathbf{T}_{\mathbf{High,Energy}}$	$\mathbf{P}_{\mathbf{Good,Fat}} \rightarrow \mathbf{T}_{\mathbf{High,Fat}}$	$\mathbf{P}_{\mathbf{Good,Salt}} \rightarrow \mathbf{T}_{\mathbf{High,Salt}}$	$\mathbf{P}_{\mathbf{Good,Sugar}} \rightarrow \mathbf{T}_{\mathbf{High,Sugar}}$	$\mathbf{P}_{\mathbf{Good,Protein}} \rightarrow \mathbf{T}_{\mathbf{High,Protein}}$
l = *Medium*	$\mathbf{P}_{\mathbf{Medium,Energy}} \rightarrow \mathbf{T}_{\mathbf{Medium,Energy}}$	$\mathbf{P}_{\mathbf{Medium,Fat}} \rightarrow \mathbf{T}_{\mathbf{Medium,Fat}}$	$\mathbf{P}_{\mathbf{Medium,Salt}} \rightarrow \mathbf{T}_{\mathbf{Medium,Salt}}$	$\mathbf{P}_{\mathbf{Medium,Sugar}} \rightarrow \mathbf{T}_{\mathbf{Medium,Sugar}}$	$\mathbf{P}_{\mathbf{Medium,Protein}} \rightarrow \mathbf{T}_{\mathbf{Medium,Protein}}$
l = *Bad*	$\mathbf{P}_{\mathbf{Bad,Energy}} \rightarrow \mathbf{T}_{\mathbf{Low,Energy}}$	$\mathbf{P}_{\mathbf{Bad,Fat}} \rightarrow \mathbf{T}_{\mathbf{Low,Fat}}$	$\mathbf{P}_{\mathbf{Bad,Salt}} \rightarrow \mathbf{T}_{\mathbf{Low,Salt}}$	$\mathbf{P}_{\mathbf{Bad,Sugar}} \rightarrow \mathbf{T}_{\mathbf{Low,Sugar}}$	$\mathbf{P}_{\mathbf{Bad,Protein}} \rightarrow \mathbf{T}_{\mathbf{Low,Protein}}$

energy, fat, salt, sugar and protein. Inspired by the three expressions of genotypes shown in Table 4.3, I design the phenotypes as having three ability levels: good, medium and bad. Hence I simplify my equation to a direct mapping $\mathbf{G} \approx \mathbf{P}$. While aiming to link this relationship to grocery data, I design the threshold table $\mathbf{T}_{\mathbf{l,n}}$ to express the strength of ability to consume each nutrient. In a real-world environment, a person's **G** is defined by the outcome pattern of a genetic test, and **P** is mapped to **G**, while **P** exhibits a direct one-to-one mapping to $\mathbf{T}_{\mathbf{l,n}}$. Such a relationship is shown in Table 4.4. The extension of adding **E** and **GE** simply seeks to create more threshold types and extend Table 4.4 and will be covered in future work.

Table 4.4 presents the mapping from **P** to $\mathbf{T}_{\mathbf{l,n}}$. In this work I treat $\mathbf{T}_{\mathbf{l,n}}$ as a matrix of variables. These variables obtain their values through optimising the fitness score of the model with the genetic algorithm. $\mathbf{T}_{\mathbf{l,n}}$ is treated as the target variables that alters my recommendations. Therefore the action of filtering different numbers of products increases or decreases the probability of choosing a product within a category. With such a relationship, I have already linked the genetic test result **G** to the product recommendations through the variation of filtering thresholds.

With the genetic and food data ready, the next step is to build a recommendation model and to optimise it with an algorithm. I explain the algorithm in the following section and apply it in my designed model in Section 4.4.

A genetic algorithm (GA) is a type of evolutionary algorithm that is inspired by the process of selection in nature. It relies on bio-inspired operations such as mutation, crossover and selection [89]. Starting from a set of population solutions, the algorithm aims to optimise a solution through changing the properties of a population set. The properties of a population are encoded in numbers representing different expressions of genes [133].

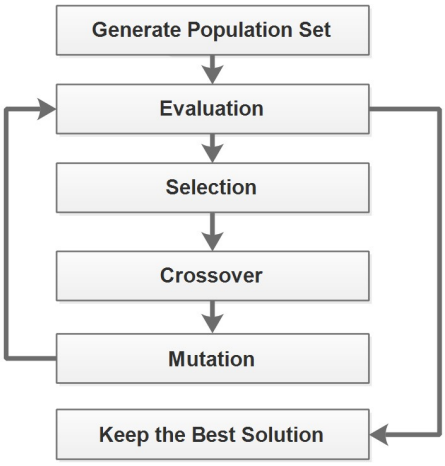


Fig. 4.5 The genetic algorithm procedure starts by generating multiple population sets. It loops through the processes of evaluation, selection, crossover and mutation. Once the training termination requirement is met, the best solution is kept.

The process of optimisation in GA is referred to as evolution. First the evolution process randomly generates individuals in a population and iterates their gene patterns with bio-inspired operations. In each iteration, the population set is called a generation. In each generation, the fitness of individuals are evaluated with the fitness scores, which are derived from predefined functions in the optimisation problem. Stochastic selections of fit individuals are chosen to proceed to the next iteration and to form a new generation [133].

The algorithm requires a few parameters to be set up before execution:

1. Population size (**Po**) is the total number of individuals to survive in each generation.
2. String length (**Len**) is the length of the genes of each individuals.
3. Probability of crossover (**Pc**) is the main control of the frequency at which the bio-inspired crossover operation take places.
4. Probability of mutation (**Pm**) is the main control of the frequency at which the bio-inspired mutation operation is executed.
5. Fitness function (**fi**) is the function to measure the fitness of individuals.
6. Termination criteria (**Te**) is used to define when the algorithm ends.

With the above parameters established, the algorithm follows the procedure shown in Figure 4.5. The first generation of the population is set up randomly using parameters **Po**

and **Len**. The generated population is evaluated by the fitness function **fi**. In the step of selection, individuals with higher fitness score are treated as the parent population. New individuals are generated with the genes of parents using the bio-inspired operation crossover and mutation with probabilities **Pc** and **Pm**. If the parents population and the generated population are sent back to the evaluation step, some individuals will be removed. The removed number of individuals is equal to the total population minus the predefined parameter **Po** [133].

4.4 Personalised Expert Recommendation System for Optimised Nutrition

In this section, I begin by defining the expression of nutrient values as variables and filtering functions , before deriving fitness score for optimisation. The input of the filter $\mathbf{p}_n^{i_c}$ is the product nutrient value, where i_c is the product identity number of the grocery category c, which varies from 1 to the number of the total product I_c of category c.

$$H_{l,n}^{c}(P_n^{i_c}) = \begin{cases} 1 & \text{if } \mathbf{p}_n^{i_c} \leq \mathbf{T}_{l,n}^{c} \quad (4.37a) \\ 0 & \text{if } \mathbf{p}_n^{i_c} > \mathbf{T}_{l,n}^{c} \quad (4.37b) \end{cases}$$

The filter function is low pass, meaning the filter outputs 1 when the threshold value $\mathbf{T}_{l,n}^{c}$ is larger than the product nutritional value $\mathbf{p}_n^{i_c}$ and outputs 0 when the opposite condition occurs.

The decision to suggest that a product is based on the function D is expressed in Equation 4.38. The output of the decision is either one or zero, which is the multiplication in the series of the outputs of filters. This procedure can be imagined as a product passing through all layers of filters and receiving a decision of one when satisfying the condition that all the nutritional factors are lower than the assigned thresholds to a person.

$$D_l(\mathbf{p}_n^{i_c}) = \prod_{n=1}^{\mathbf{M}} H_{l,n}^{C}(\mathbf{p}_n^{i_c}) \tag{4.38}$$

Scanning through all the products, the number of suggested products within a category is expressed in Equation 4.39, which is the sum of all the $D_l(\mathbf{p}_n^{i_c})$ of products within category c.

$$s_l^c = \sum_{i_c=1}^{I_c} D_l(\mathbf{p}_n^{i_c}) \tag{4.39}$$

Now the average nutritional value of the suggested products $a_1^{n,l}$, can be obtained using Equation 4.40. The average of all nutritional value of all products a_2^n is obtained using Equation 4.41

$$a_1^{n,l} = \sum_{i_c=1}^{I_c} \frac{\mathbf{p}_n^{i_c} \odot D(\mathbf{p}_n^{i_c})}{s_l^c} \tag{4.40}$$

$$a_2^n = \sum_{i_c=1}^{I_c} \frac{\mathbf{p}_n^{i_c}}{I_c} \tag{4.41}$$

In order to achieve more realistic experiments, a simple behaviour model with a Gaussian function is introduced. This model is used to describe the probability of a person following the suggestion given by the system, as shown in Equation 4.42. It is assumed that the probability of the person following the suggestion is at its maximum value when half of the product is recommended. When the number of product recommendations exceeds 50% of the total products, the probability decreases due to the psychological doubt that the system does not filter any products for the person. On the other hand, the probability also decreases when the number of the products recommendations is below 50%, owing to the frustration of having fewer options. Note that this assumption of behaviour can be changed to fit the real-world data if a survey is conducted on the topic.

$$Q_l^c = \mathrm{f}(\frac{s_l^c}{I_c}) = 1 - 0.5 \times e^{\frac{(\frac{s_l^c}{I_c} 0.5)^2}{4.6^2}} \tag{4.42}$$

Finally, I compare two scenarios. The first scenario is the sum of the consumption of the nutrients with the product recommendation and the consumption of the nutrients without the recommendation from the system. This can be obtained through the sum of the probability of following the suggestion multiplied by the average nutritional value of the suggested products and the remaining probability multiplied by the average nutritional value of all products. The second scenario is simply the average nutritional values of all products, as shown in Equation 4.43 and 4.44.

$$v_1^{n,l} = a_1^{n,l} \times Q_l^c + a_2^n \times (1 - Q_l^c) \tag{4.43}$$

$$v_2^n = a_2^n \tag{4.44}$$

The difference between the two scenarios is the reduced consumption $r_{n,l}$ of nutrition n with the nutrition threshold boundary l, as shown in Equations 4.45 and 4.46.

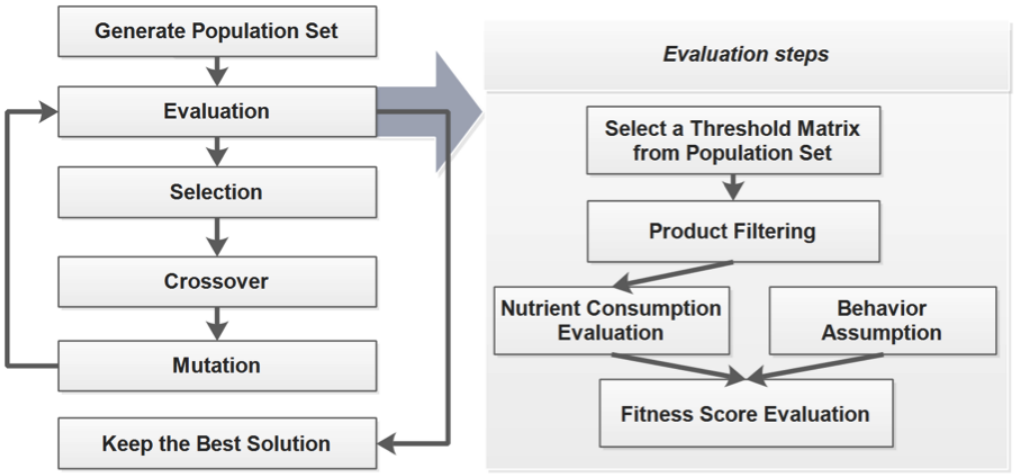


Fig. 4.6 The procedure of recommendation optimisation with a genetic algorithm. In my design, the evaluation steps are extended to perform product filtering, nutrient consumption evaluation and fitness evaluation based on behaviour assumptions.

$$r_{n,l} = v_2^n - v_1^{n,l} \tag{4.45}$$

$$r_n = \sum_{l=1}^{L} \frac{r_{n,l}}{L} \tag{4.46}$$

With the aim of considering all nutrients, for $n = 1, 2, \ldots, 5$, or in other words, energy, fat, salt, sugar and protein, the fitness score f is measured based on the sum of the normalised form of the reduced consumption of nutrition z_n, as shown in Equations 4.47 and 4.48.

$$z_n = \frac{r_n}{a_2^n} \tag{4.47}$$

$$fi = \sum_{n=1}^{M} z_n \tag{4.48}$$

The fitness score fi is designed as a measurement of the difference in nutritional intake average after receiving the recommendation, with the aim of maximising this score. GA is applied due to its outstanding performance for stochastic optimisation. Indeed, it is inspired by the process of natural selection, which is commonly applied to generate high-quality optimisation solutions. There are three main bio-inspired operations: mutation, crossover and selection.

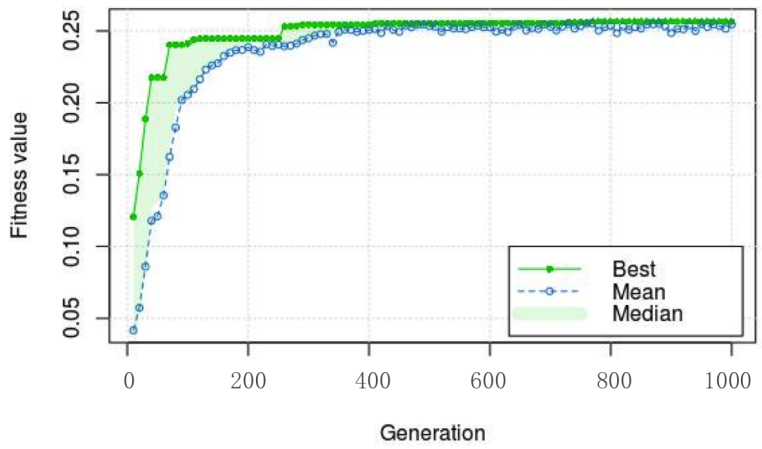


Fig. 4.7 Optimisation of the fitness score using a genetic algorithm. The best value is found at iteration 410.

The procedure of recommendation optimisation using the GA is shown on the left side of Figure 4.6. In the step of population generation, random sets of threshold matrices are initialised. The generated threshold matrices are fed one by one to the evaluation step, where the proposed recommendation model using Equations 4.37 to 4.47 is applied as shown on the right side of Figure 4.6.

The evaluation steps include selecting one of the sample threshold matrices from the population sets, product filtering with Equations 4.37 and 4.38, nutrient consumption evaluation following Equationss 4.39 to 4.41 and fitness score evaluation with the behavioural assumption described in Equations 4.42 to 4.48. Once the fitness scores are evaluated for all threshold matrices in the population sets, they are compared and the best few sets of the threshold matrices that generate higher fitness scores are selected. The non-selected threshold matrices are deleted from the population sets.

In the step of crossover, random pairs of the selected threshold matrices are formed and some parameters of randomly selected locations are swapped between pairs. Furthermore, some locations of the selected sets of threshold matrices are modified with a low probability in the step of mutation. Finally, the new modified population sets of threshold matrices are sent back to the evaluation step.

In this work, package "GA" is applied to achieve optimisation tasks [116, 115]. GA is set up as the algorithm to maximise the fitness score f of the model with the input variable $\mathbf{T}^C_{l,n}$ of grocery category C under the constrain $\mathbf{T}^C_{1,n} > \mathbf{T}^C_{2,n} > \mathbf{T}^C_{3,n}$, or in other words $\mathbf{T}^C_{High,n} > \mathbf{T}^C_{Medium,n} > \mathbf{T}^C_{Low,n}$.

The categorised data is fed into the decision recommendation models and the genetic algorithm is set up to maximise the fitness score with 1,000 iterations of the three databases.

Table 4.5 Average reduction of nutrient consumption considering per 100g of the products as a result of following the system recommendation compared to random decisions.

Per 100g of the product	Biscuit& Cereal& Cookie	Chocolate	Yogurt	Rice& Noodle& Pasta
Energy	61.2kJ	68kJ	63kJ	200kJ
Fat	3.5g	1.64g	1.58g	0.38g
Sugar	5g	5.6g	5.82g	0.3g
Salt	0.101g	0.29g	0.07g	0.1g
Protein	1.12g	1.16g	1.18g	0.5g

Figure 4.7 shows the optimisation of fitness score where the value saturated at iteration 410 and the maximum fitness score remains at 0.2568. It is assumed that the decision of product consumption is at the same weight: 100 g. The improvement is measured through the reduction in consumption of a nutrient, compared between scenarios of a decision made with and without the personalised suggestion. It is found that the reductions of nutrition intake do not show significant differences between databases, although the values between different categories are comparable. In Table 4.5, the improvement for each nutrient from different categories is the average of the experiment outcome based on the three databases.

The category rice, noodle and pasta reveals a considerable reduction in energy intake, whereas fat, sugar and protein exhibit a smaller reduction, as most of these products either contain a low value or none of these nutrients. The category chocolate yields greater improvement on salt, while the category biscuits, cereal and cookie presents a greater improvement in fat. The possible key nutrition that causes difference in diets can be presented through relatively greater improvements as described above.

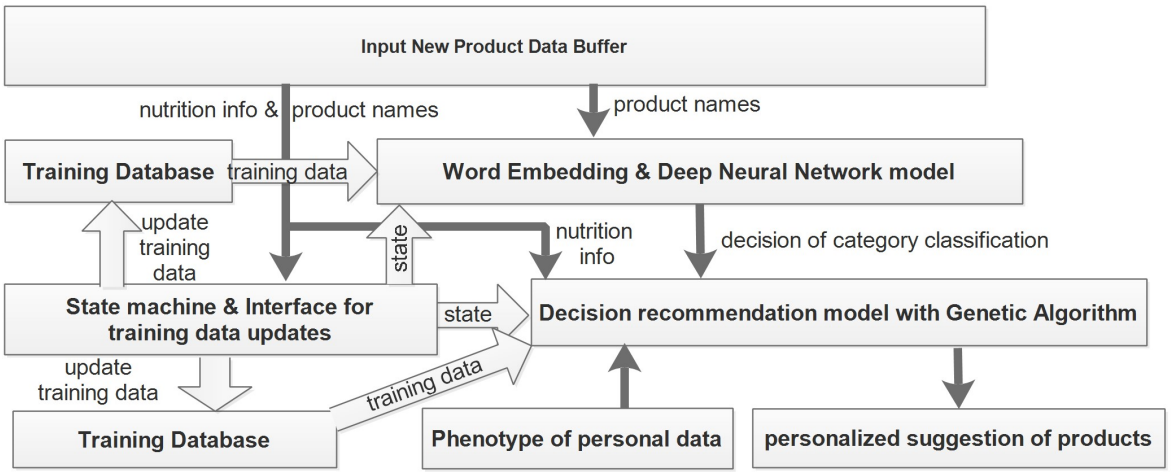


Fig. 4.8 The architecture of the Personalised Expert Recommendation System for Optimised Nutrition (PERSON). This framework is controlled by a state machine setting up actions from the deep neural network and the decision recommendation model. The two models are optimised using training data from a database. The deep neural network categorises new data from the new data input buffer and the decision recommendation model performs a personalised suggestion service based on personal phenotype data.

4.5 Framework for Application

In this section, I explain the two main models applied in my proposed expert recommendation system and describe the framework architecture, operation, and application of the Personalised Expert Recommendation System for Optimised Nutrition, denoted as PERSON. The framework combines different components, which operate as one system. Later in this section I explain how PERSON operates as a service to compare foods and to provide suggestions.

The top level block diagram of the system is shown in Figure 4.8. The whole system operates in two states: a training state and a recommendation provider state. Each block individually operates as:

1. Input New Product Data Buffer: the top of the figure shows the input data buffer, which temporarily stores the new product information.

2. State Machine & Interface for training data updates: the block receives new product information including the nutrition fact tables and the product names. The training data is selected and updated with an operation of human experts through an interface. The state machine switches to the training state during the the training updates and sends the state signal to all other blocks.

3. Training Database: the block receives updates from the state machine block and stores the updated data based on expert-determined categories.

4. Word Embedding & DNN model block: the block is controlled by the state machine. It receives data from the training database during the state of training and performs category classification to the new product names with the Deep Long Short Term Memory(DLSTM) model when the state is switched to the recommendation provider.

5. Decision recommendation Model: this block also operates based on the state signal, which receives training data and sets up the nutrition threshold matrix using the GA introduced in Section 4.4. In its state as recommendation provider, the decision recommendation model provides a personalised suggestion of products based on the phenotype of the personal data, the category classification outcome of the DNN model and the corresponding nutrition facts information. The output is a list of filtered products based on a set of threshold $T_{l,n}$ that maps to the phenotype $P_{l,n}$.

During operations for application, lists of grocery products are constantly updated to the system through the input new product data buffer.The products are then categorised into groups by the Word Embedding & DNN model block. Product filtering and recommendation processes are carried out in the decision recommendation model block. Each consumer can receive a personalised product recommendation list by uploading their Phenotype/Genotype datasets from the decision recommendation model. Note that the Word Embedding & DNN model block and the decision recommendation model can be retrained to fit the new grocery data with higher accuracy of categorisation by switching the state and manually adding labels to the new product data.

Considering the flow of service processing I show a top level system block diagram in Figure 4.9. On the left side, two types of input are separated, “Server & Database” which covers the data pre-processing procedure and blocks within the “User-End”, or the procedure or action by which the user uploads his or her DNA information and barcode data.

There are three sources for the three types of data collection. The supermarket grocery data is collected from the online shopping website, which was previously introduced in Section 3.1. The Genetic-Nutrient mapping data is directly provided by the geneticist at this stage.

The collected supermarket grocery data is processed through data curation and analysis. All the preprocessed data is stored in a database, which responds to a request of service by customers. The service includes “Direct To Product-DNA Personalised Mapping” and “Suggestion Delivery”.

The aim of this stage is to map the product data to DNA data, while the product is mapped to its barcode. Consumers receive a suggestion whenever a barcode is selected or scanned based

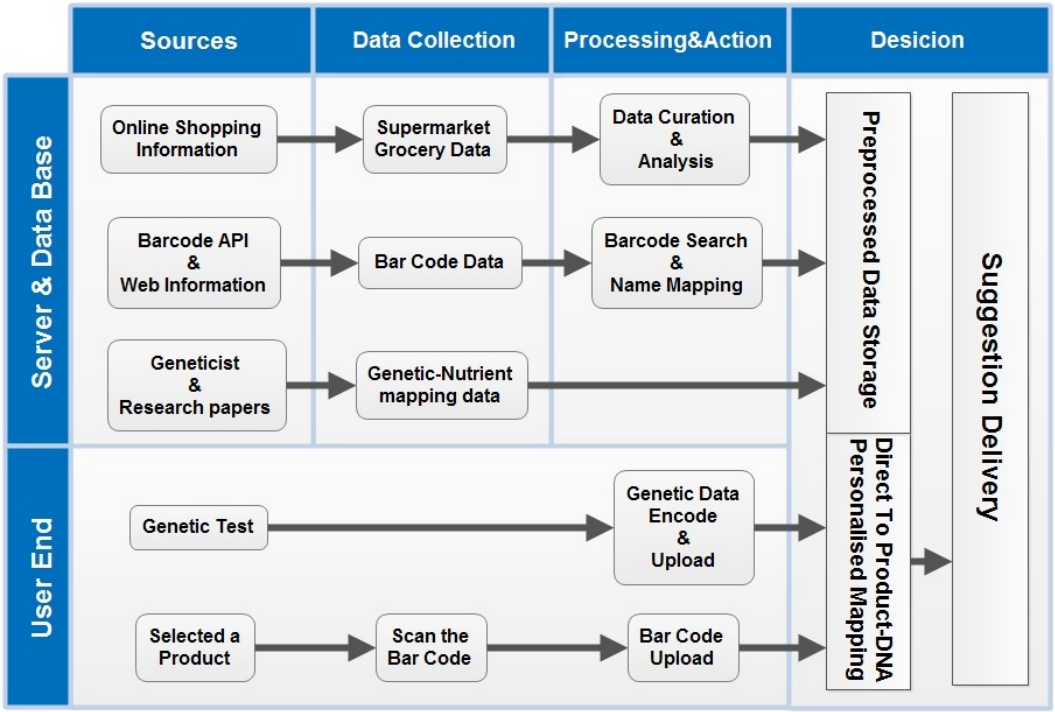


Fig. 4.9 Top-level block diagram for the service system. There are two sides to service processes. One is related to the server and databases in DNAnudge. It involves data such as product information, bar-codes and genetic mapping. The other side is related to the user and takes the individual genetic test data and the product that the user selects at the time of shopping.

on their DNA pattern. An example is shown in Figure 4.10. The genetic results of this stage are confirmed by biological researchers. A threshold of each nutritional factor is predefined by a biologist. DNA data is selected based on the chosen nutritional factors. In this example, energy is selected as the nutritional factor and a gene with the name "APXX" is chosen to observe the health effect. SNP is the key cause of difference in the whole gene, for which the "rs50xx" on "APXX" is selected. For one SNP, three patterns can usually be found, and a threshold value that represents the boundary of acceptance of the amount of nutrients is mapped to these three patterns. The threshold of a particular customer is passed to the final stage for comparison when his or her DNA patterns are uploaded.

On the other hand, whenever a product is selected by a consumer, the barcode is scanned and uploaded to system, as shown on the right side of Figure 4.10. UPC and EAN barcodes are generated by the barcode scanner. Furthermore the name of the product mapped to the number is found. The nutrient value is extracted from the nutrition facts table based on the selected nutrition fact. This value is compared with the threshold. With the example product shown in Figure 4.10, it is rejected due to the fact that the value of the energy is higher than the energy

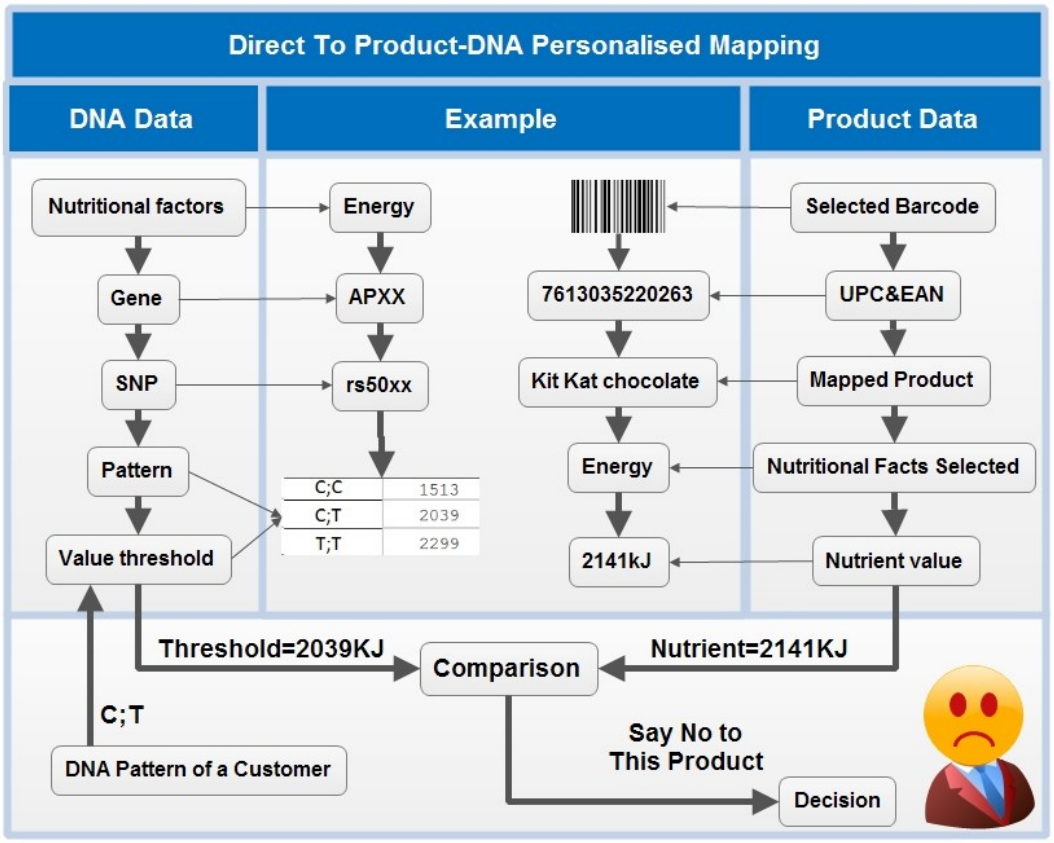


Fig. 4.10 This is an example of the PERSON framework in service operation. The customer completed a genetic test and picked up a product called a Kit Kat bar™. The gene-mapping data comes from the top left of the figure, the customer's gene information drops in from the left button and the two pieces of information are used to generate a threshold. The threshold filters out the product by examining the product information from the top right.

threshold. The comparison procedure works in a similar way to a low pass filter in the case of observing the nutrition fact "energy".

With this scanning barcode approach, I ensure ease of implementation via existing hardware such as smart mobile phone applications. This is because barcode scanning is a longstanding technology and is frequently available. In reality, the suggestions regarding all products to a pattern of genes are generated when consumers upload their genetic test results. The generated results are loaded to a device that acts as a mapping dictionary, as shown in Figure 4.10.

The aim of this technology is to instantly discover which products are the best for each consumer's unique genetic pattern. It helps consumers to make informed choices. These informed decisions are inspired by the concept of Nudge theory, introduced in Section 2.1. Every time when a person uses this application, it is an nudge step that optimises his or her decisions, and presents an opportunity to develop a healthier self.

Chapter 5

Case Study: Supermarket Trial for DNAnudge

Nudge is a concept that aims to generate indirect suggestions to influence an individual's behaviours. In order to study the influence of Nudge on food shopping decision-making, a trial was conducted with a leading supermarket in London. In this chapter, I present a model for this trial and analyse the data I obtained.

Two new Nudge concepts have been created by Professor Christofer Toumazou and Dr Maria Karvela: 1) **Nudgeomic (self-nudge):** describes the self-nudge process. This concept aims to explain variations in individual behaviours that have no influence from the outside. 2) **DnaNudge:** a nudge process that seeks to alter an individual's behaviour based on their genes. In this trial, a DnaNudge occurs when recommendations are obtained based on genes after receiving supermarket product information from each food category.

A top-level process is shown in Figure 5.1. All products in each category are assigned to one label: recommended or not recommended. The recommended products are marked as green and the variation in the green products purchased by customers throughout the trial is shown in Figure 5.2. I collected the retail data of both (the year before the trial) and (the year of the trial) from the supermarket. Due to the fact that the average number of recommended products purchased for each customer varies significantly compared with non-recommended products, I focused on analysing the purchase recommended products (green products). The overlapping number of this year and last year is defined as regular purchase, whereas the difference is the nudged purchase. The nudged purchase are due to the influences of Self-nudge and DNAnudge.

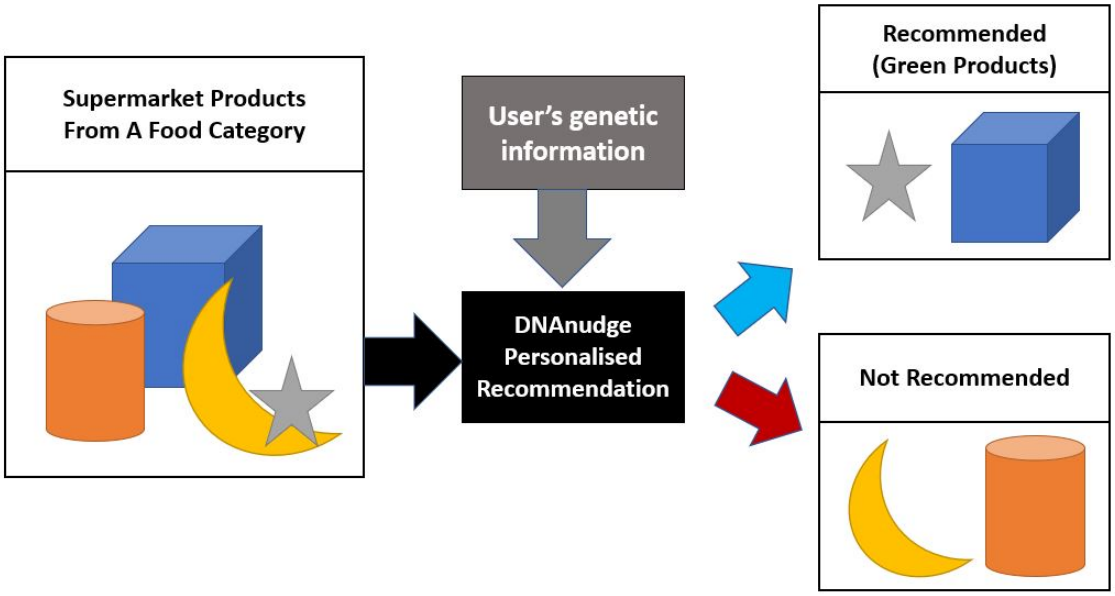


Fig. 5.1 In a supermarket trial, DNAnudge generated recommendations based on a user's genetic information. The recommendations were divided into two groups: recommended (green products) and not recommended products.

5.1 Trial Setup

In this trial, customer data was collected from loyalty cards: customers use loyalty cards when they pay their bill at the till. These collected data include the types of products they purchased, the money they spent on products, the number of products they purchased and more. The targeted participants were separated into two groups, as shown in Figure 5.3: 1) **Target group:** customers who receive DNAnudge recommendations when they scan the barcode of food and drink products using an application (app) on their mobile device. The usages of the app are tracked in the cloud to study the correlation between usages and behavioural variations. 2) **Reference group (Ref group):** customers who are part of the trial but do not have any usage record of DNAnudge service. Therefore I assumed that they do regular shopping freely without receiving personalised recommendations.

In the year before the trial, both the reference group and the target group shopped without receiving personalised recommendations. The reference group did not receive recommendations during the trial, while the target group received recommendations from DNAnudge. This process is shown in Figure 5.4.

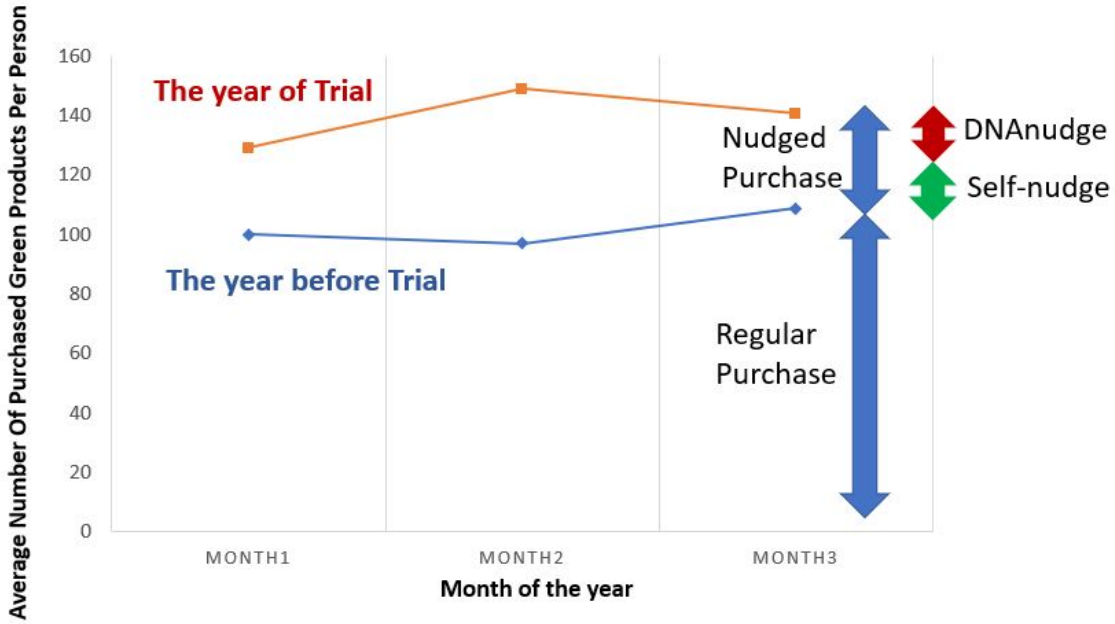


Fig. 5.2 The average number of purchased green products per person versus the month of the year. There are two sets of data: one from the year before the trial and the other from the year of the trial. The purchase difference of the two data sets is considered the purchases that were influenced by the nudge effect, consisting DNAnudge and Self-nudge effects.

5.2 DNAnudge Trial Outcome

In this section, I describe a mathematical model to calculate the influences of personalised recommendations on shopping purchases and confirm my designed model based on the real-world data collected.

I compared the loyalty card data from the same time period for this year and last year, for example 2019 Jan versus 2018 Jan. For the year before the trial, both groups shop without receiving any recommendations from DNAnudge. Therefore their shopping behaviours were assumed to be influenced only by "self-nudge". During the trial, the reference group did not receive any recommendations from DNAnudge, while the target group received personalised recommendations from it. Thus the target group's shopping behaviours were assumed to be influenced not only by self-nudge but also by DNAnudge. Figure 5.5 presents a detailed comparison between the reference group and the target group.

The sum of all effects on the target group is defined as $T(y,m)$ while the sum of all effects on the reference group was defined as $R(y,m)$, where $y = 0$ represents the year before the trial and $y = 1$ represents the year of the trial, $\forall m \in 1,2,3$ represents the month of the trial. $T(y,m)$

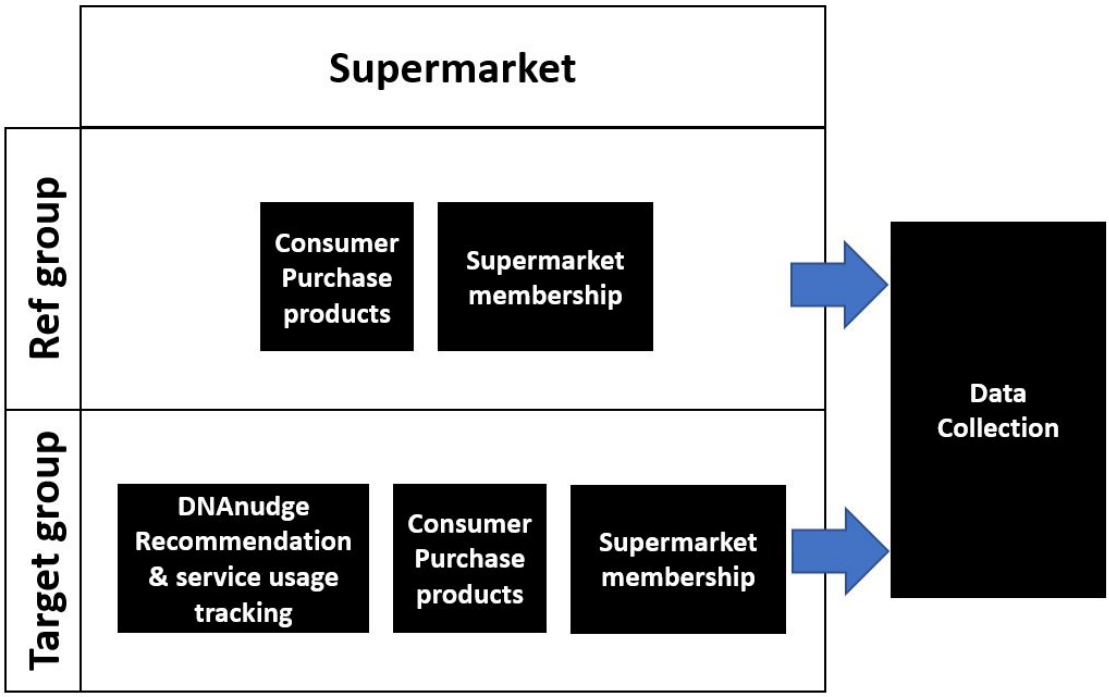


Fig. 5.3 In this supermarket trial, I collected consumer purchase records and membership identification numbers from the reference group. From the target group, apart from consumer purchase records and membership identification numbers, I also kept track of the DNAnudge service usage record and the recommendations given to the user.

can be expressed as the sum of the effect of Self-nudge ($TS(y,m)$) and the effect of DNAnudge ($TD(y,m)$) as shown in Equation 5.1.

$$T(y,m) = TS(y,m) + TD(y,m) \tag{5.1}$$

Self-nudge was assumed to affect personal bias monthly and yearly. The difference of self-nudge between each year was defined as $Y(y,y-1)$, which is the variation of environment between each year. Monthly personal effects can be expressed as $MP(m) = M(m) * P(m)$, where $M(m)$ is the difference in Self-nudge between different month and $P(m)$ is the difference of self-nudge between each group. These are used to represent the seasonal variation and personal bias of self-nudge. The monthly personal effects $MP(m)$ can also be expressed as in Equation 5.2. Assuming $YR(1,0) = YT(1,0)$ and $TS(1,m) = R(1,m) * MP(m)$. From equation 5.1, the DnaNudge effect can be expressed as $TD(1,m) = T(1,m) - TS(1,m)$, where $T(1,m)$ and $TS(1,m)$ are known.

$$MP(m) = RS(y,m)/TS(y,m) \tag{5.2}$$

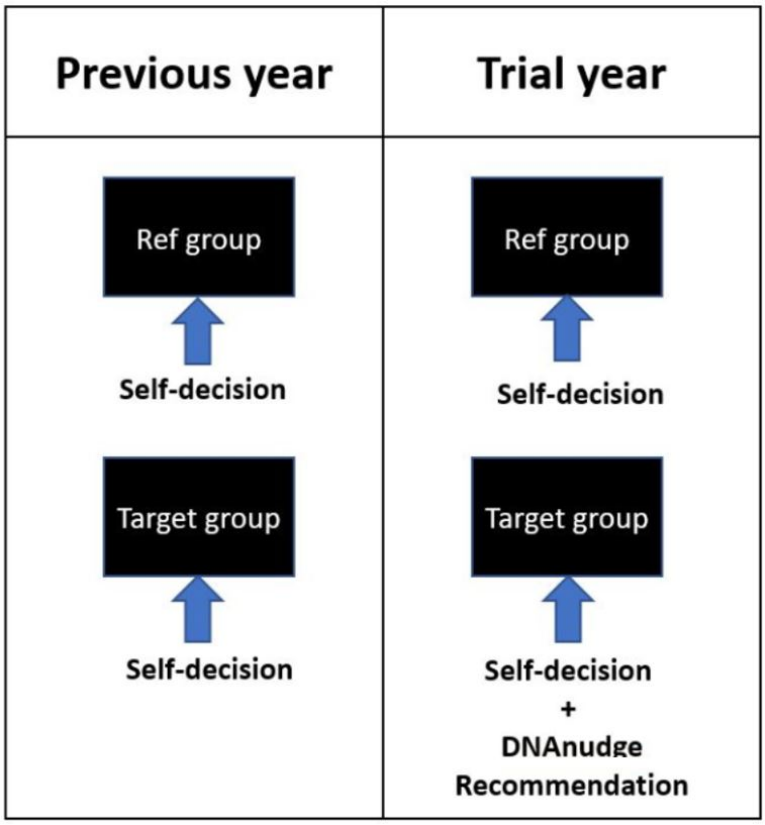


Fig. 5.4 Considering the experiment setup and relevant influence, only the target group from the trial year performed self-decision with DNAnudge recommendations.

The DnaNudge effect $TD(1,m)$ can be further expressed with immediate effect $TI(m)$ and memory effect $TM(m)$ as shown in Equation 5.3. The definition of immediate effect and memory effect will be explained later.

$$TD(1,m) = TI(m) + TM(m) \tag{5.3}$$

With the aim of exploring the relationship between service usage and purchase increase, I separated the target group into three smaller groups based on the frequency they use our mobile application to receive recommendations: 1) **High use group:** customers who used DnaNudge service at least once a week. 2) **Medium use group:** customers who used DnaNudge service at least once every two weeks. 3) **Low use group:** customers who used DnaNudge service at least once a month. To present the different smaller groups, Equation 5.3 can be re-written as in Equation 5.4, where g is used to represent different smaller groups.

$$TD(1,m,g) = TI(m,g) + TM(m,g) \tag{5.4}$$

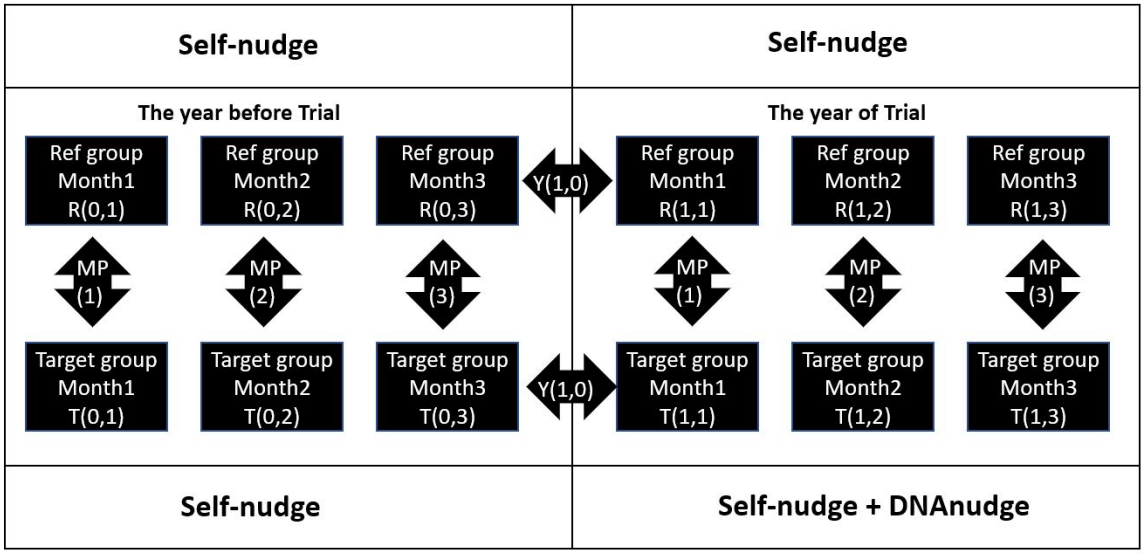


Fig. 5.5 Relationships between the reference group and the target group are expressed in variables. Three months of data from the year before the trial and the three months of data from the year of the trial are shown.

Figure 5.6 shows the average weekly usage of shopping with DnaNudge service per person from real-world data. The high usage group presented the highest weekly usage throughout the three months of the trial, while the medium-usage group has less weekly usage and more than the low usage group. The weekly usage for all three groups dropped gradually during the three months. In particular, the high usage group dropped significant at the third month, probably due to the fact that the customers had become more familiar with the options appropriate to their needs.

For comparison purposes, I focus on the green products purchased by customers during the trial. First, as shown in Figure 5.7, the total number of green products purchased for all three groups did not present an obvious trace. Neither do the self-nudged purchasing and DnaNudged purchasing. However, after I further investigating the immediate effect and the memory effect, I became able to identify some interesting results.

The immediate effect of DNAnudge for each group $TI(m,g)$ can be expressed using Equation 5.5. $U(m,g)$ is the recorded shopping data from the month. $W(g)$ is the weight for each group or the group slope for immediate effect, which can be calculated using $TI(m,g)$ and $U(m,g)$ from real-world data.

$$TI(m,g) = U(m,g) * W(g) \tag{5.5}$$

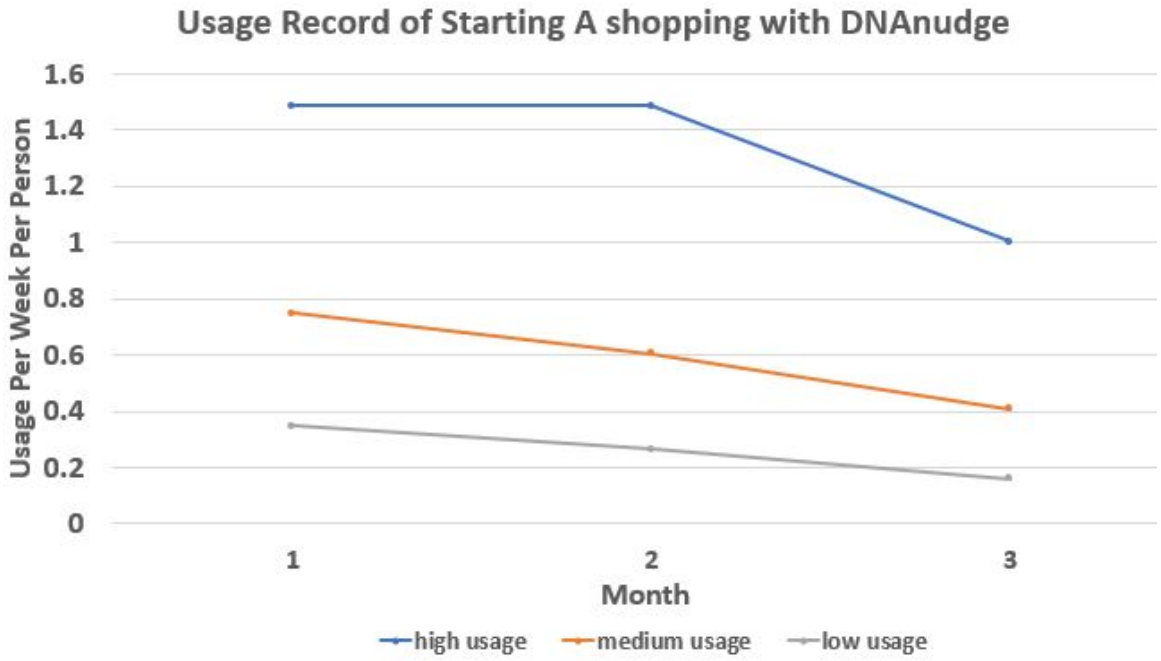


Fig. 5.6 Usage per week per person versus the month of trial is presented. The users were split into three groups, where the high usage group used the DNAnudge app at least once every week, the medium usage group used the DNAnudge app at least once every two weeks and the low usage group used the app at least once every month.

Figure 5.8 shows the immediate effects of $TI(m,g)$ on the number of green products purchased by three groups. The lower usage group had the highest impact when it increased its usage of DNAnudge service more. This scenario may be because high usage customers are easily exhausted by too many recommendations and information whereas other customers were surprised by new information. Alternatively, it may be because the decisions made by high usage customers were already good and were therefore difficult to improve using DnaNudge.

Another important factor is the memory effect $TM(m,g)$, which is expressed in Equation 5.6. It can be further extended into a series of sums of immediate effects multiplied by a forget function $F(x)$, where x is the month distance from the current month, as shown in Equation 5.7. The memory effect only features when customers receive recommendations from the DNAnudge service after one month. Assuming $F(0) = 1$, Equation 5.7 can be re-written into Equation 5.8. The forget function can be obtained from real-world data and be used to represent the time required for customers to forget the DNAnudge recommendations.

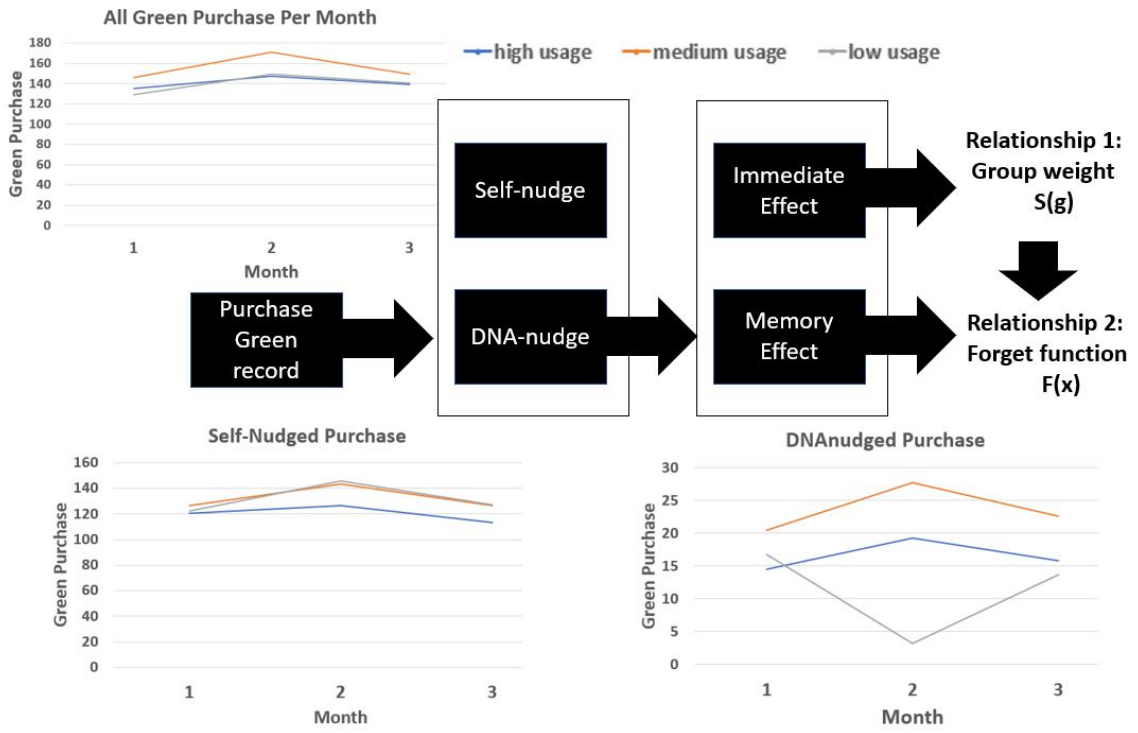


Fig. 5.7 Consumer purchasing of green products is assumed to be driven by self-nudge and DNAnudge effects. The DNAnudge effect is further split into immediate and memory DNAnudge effects.

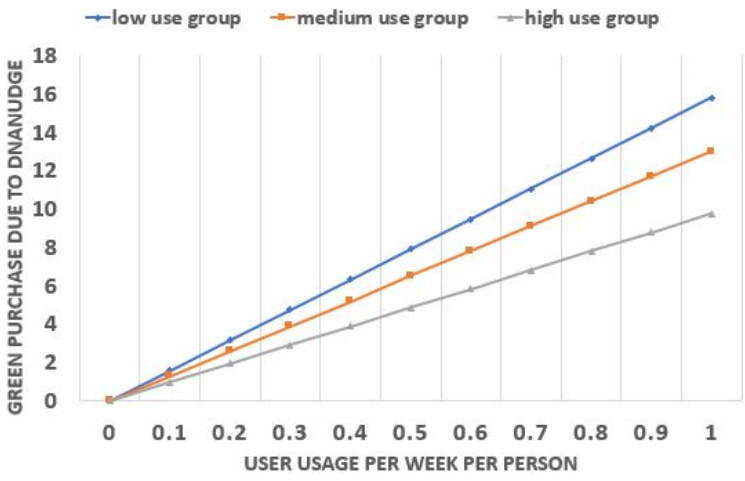


Fig. 5.8 Green purchases due to DNAnudge versus user usage per week per person. The low usage group shows a steeper slope, which means the consumers from the low use group are easier to influence with DNAnudge recommendations. In comparison, the consumers from the high use group are the hardest to influence.

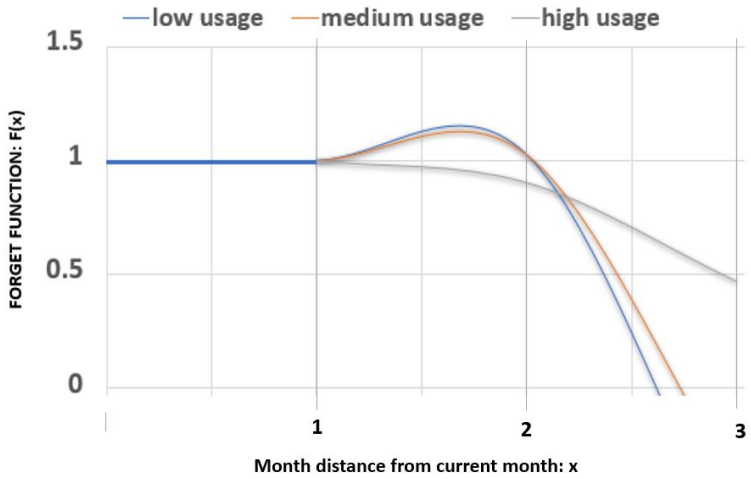


Fig. 5.9 The value of the forget function of the memory effect versus the distance to the current month. The space before 1 on the horizontal axis represents the current month and, therefore, remains at 1 (no memory effect). Consumers in the low and medium DNAnudge usage groups reach 0 for the forget function in the space between 2 to 3 months of the time distance to the current time. This indicates that consumers from the two groups forget the learned recommendation in 2 to 3 months.

$$TD(1,m,g) = TI(m,g) + TM(m,g) = TI(m,g) + F(1) * TI(m-1,g) + F(2) * TI(m-2,g)... \tag{5.6}$$

$$TM(m,g) = \sum_{x=1}^{m-1} F(x) * TI(m-x,g) \quad for \quad m > 1 \tag{5.7}$$

$$TD(1,m,g) = F(0) * TI(m,g) + TM(m,g) = \sum_{x=0}^{m-1} F(x) * TI(m-x,g) \tag{5.8}$$

Assuming that the forget function is a polynomial regression, Figure 5.9 shows the memory effects $TM(m,g)$ on green products purchased by three groups. The low and medium usage groups have a sudden high memory effect when they have just learned a new recommendation. However they forget quickly after one month. The high usage group tends to remember a new recommendation longer. These results may be explained by the high usage group caring more about the learned health information.

In this chapter, I confirm the existence of the self-nudge effect and the advantages of the DNAnudge effects. In order to construct a mathematical model of the DNAnudge effects, I have defined an immediate effect and a memory effect of DNAnudge. By analysing real-world

supermarket data, I can conclude that the high usage of the DNAnudge app helps people to remember the DNAnudge recommendations longer and results in more healthy decisions. On the other hand, the lower usage of DNAnudge customers tend to be easily influenced by the DNAnudge recommendations but forget much faster. This analytic model can potentially help manufactures and supermarkets to produce healthier products. It will also aid in bring my designed framework PERSON closer to reality.

Chapter 6

Grocery Business Unionism Strategy in Neural Expert System Simulation

Establishing the framework of the Personalised Expert Recommendation System for Optimised Nutrition and providing a commercial service represents the first step toward achieving DNAnudge. This first step aims to nudge the consumer towards better shopping decisions. Once this service gathers enough users and starts making an impact on the market, the design of the service can move on to the discussion of nudge manufacturers and retailers. In this chapter, I begin with a discussion of the general concept of how data can become valuable through a data value chain and describe the background of behavioural simulation with game theory and reinforcement learning in Section 6.1. I apply the proposed PERSON to build a simulation model that mimics the interactions between manufacturers and consumers in section 6.2. The simulation model is denoted as G-BUSiNESS. In section 6.3, I conduct experiments considering the concept of Business Unionism, where I treat all manufacturers as one union – a food category. All members of this union seek to increase or protect their immediate economic interests.

6.1 Background of Behavioural Simulation with Reinforcement Learning

Nowadays data comes from all imaginable sources in real time, while good quality analytic outcomes are on demand [87]. Companies recognise the importance of data processing due to the problems occurred along with the expansion of data volume, velocity and variety (3V). In order to overcome such difficulties, data value chain was invented to capture, organise, select key data and make decisions based on observation [64]. The key to understand data processing

and handling problems is to take a thorough examination of each step of the data life-cycle. The concept of data value chain has been accepted by many research teams and even implemented as a framework to overcome big data problems [129]. Numerous programming tools in R and Hadoop have also been created to address big data problems with different techniques, such as "Map and Reduce" [110].

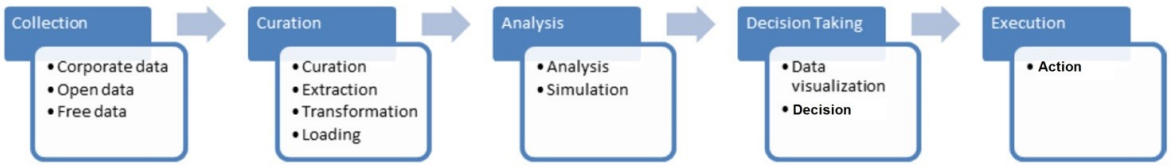


Fig. 6.1 The data value chain process consists of collection, curation, analysis, decision and execution.

As shown in Figure 6.1, a typical data value chain starts with collecting data from three sources: Corporate data, Open data and Free data. Corporate data have high qualities and usually are protected from the public by companies. Open data have medium qualities that can be freely accessed, such as websites and news. While free data is freely available for anyone online with no restrictions. Once the data is collected, based on the three different quality levels, different pre-processing steps are required, including extraction, cleaning, transformation, normalisation and loading. These steps can be referred to as data curation. After the data has been curated, analysis and simulation can be employed and the analytic results are presented through visualisation for decision making. Once a decision is made, the action of the decision may influence the environment of the sources where the data is collected, resulting in the collection of new data [87]. This chain process aims to increase the values of a company or a service between each cycle through influencing the environment. Within each cycle the value of data increases following each step of the process while the execution of an action is worth the most.

Data Value Chain is widely used to provide a general direction when designing a data driven business process. To apply it in practical uses, it requires detail consideration and evaluation of the values created by each step and each cycle. In the next section, I propose a potential way to look at the influence of the framework PERSON in the food industry from a general perspective.

The concept of Nudge aims to remind people to take better decision through various applications. Looking at the influence with measurement is difficult because following recommendations involves highly complex reasoning.

Given my interest in how consumers react to recommendations of products with better nutrition, the understanding of consumer behaviour and simulation via game theory is my next

objective. Furthermore I aim to implement a reinforcement learning Artificial Intelligence (AI) agent to perform behavioural optimisation. The agent will be designed to capture better decisions to influence the decision making of consumers and product suppliers.

As shown in Figure 6.2, an overview of the simulation framework can be divided into three stages: 1) Consumer behaviour study, 2) Game theory modelling of trade, 3) Reinforcement learning AI agent. The relationship between the three topics can be described as a board game. The consumer behaviour simulation is equivalent to the board and chess, game theory modelling refers to how the game is played, and the reinforcement learning AI agent is the operator who controls players' actions. In my behavioural study, I aim to collect and simulate evidences of how consumers in reacting to products and potentially extend to the strategies of retailers and suppliers react to purchase behaviour variations. More importantly I strive to cover the influence of decisions with an optimised recommendation. Once the evidence of behaviours is collected, a framework using game theory is then developed to model scenarios, discover hidden relationships and find available strategies between players, consumers, retailers and suppliers. Finally, utilising the simulated environment, an artificial intelligent agent with reinforcement learning techniques is developed to capture relations and generate optimised responses. Existing work about game theory frameworks with reinforcement learning focuses on demonstrating reinforcement learning based solutions on supply chain management and retailer pricing and ordering [107, 135, 36]. However, they do not cover behavioural studies and are mainly based on assumptions.

The goal of this research is to achieve a realistic simulation of the behavioural environment, generate game models to capture important factors and create agents to adapt or interact with the environment. Behaviour represents actions and mannerisms that are carried out by individuals, organisms or systems within particular environments. It is a response to internal or external, conscious or subconscious stimuli or inputs [88]. From a behaviour informatics perspective, a behaviour can be represented as a behaviour vector [18]. The process of reacting to products or services is referred as consumer behaviour. It involves consumption, a process of go through and purchasing [127]. Begining with recognising their own requirements, consumers go through products, while considering different aspects including types of products, available budgets, frequency of consumption and other impact factors [127]. Two types of impact factors can influence consumer decisions: internal and external. The former includes attitudes, needs, motivations, preferences and perceptual processes while the latter comprises marketing activities, social and economic factors, and cultural aspects [127].

A model focusing on the decision making process regarding consumer behaviours has been proposed by Lars Perner [100]. It begins with recognising needs and searching for relevant information. Once the target is set, a consideration of high or low involvement is then made. If

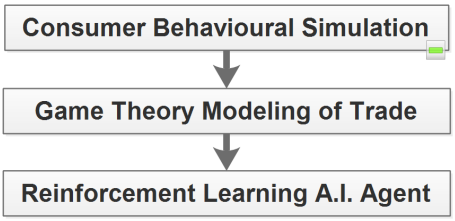


Fig. 6.2 The proposed simulation framework starts from building a consumer behavioural simulation, applying the simulation in a game theory set up and using a reinforcement learning AI agent for decision optimisation.

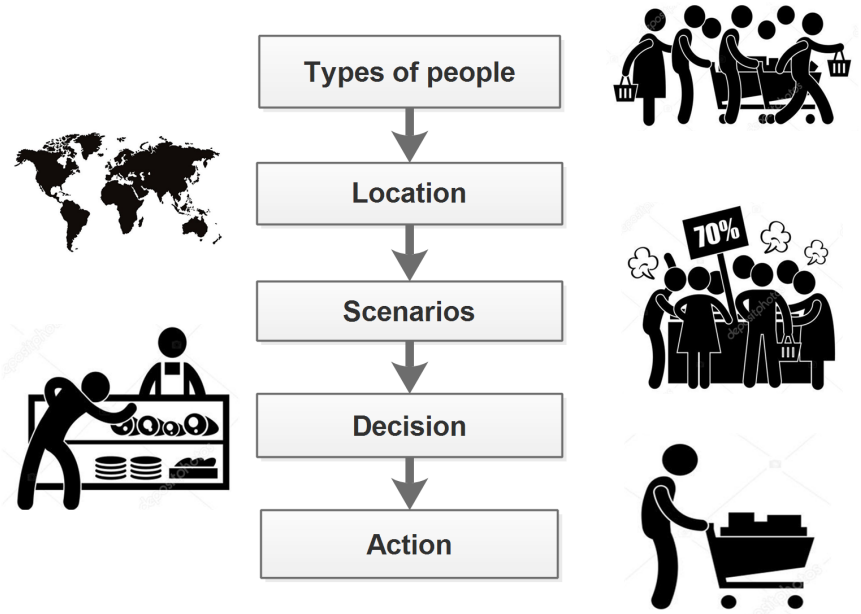


Fig. 6.3 Components of the consumer behavioural simulation, including types of people, location, scenarios, decision and action [102, 72].

the target product has low involvement, the decision to purchase alternatives is only based on money. As for a target product with high involvement, the decision process covers a greater range of factors, such as reading reviews, asking relatives, or comparing prices, qualities and trade-offs.

Clemons proposed a marketing tool called 4P standing for Price, Promotion, Product and Place [29]. In the business world, product prices are usually set up similar to competitors while making profits. When the price is high, consumers tend to purchase less and use the product longer. Promotion is believed to have a direct correlation with advertisements. However, nowadays some businesses achieve success with products despite little or no advertising [29]. This can be due to word of mouth from consumers using social media. The product also

influences consumers' behaviour through willingness to pay and preference. For example, it is difficult to force a Coca-cola fan to purchase Pepsi, even though Pepsi is cheaper. The placement of product can be less influential due to the plenty availability of goods online [29].

As shown in Figure 6.3, my draft plan is to first gather evidence related to my study from surveys, articles and other available data. The evidence will be summarised as a process starting from choosing one type of people, location, scenarios, decisions and actions. The process is used to build a simulation and is then established as the background of game theory.

Game theory is a study of conflict and cooperation between rational decision-makers, which is used in economics, political science, and psychology, as well as in logic, computer science and biology [92]. The theory starts from a zero-sum game, in which one person's gains are the other person's losses. Later, game theory has since been extended to explain a wide range of behavioural relations. The history of this theory can be traced back to the 1930s, developed extensively in the 1950s and applied to biology in 1970s [120]. Some well known applications were proposed by the Nobel Memorial Prize holder, Jean Tirole, in Economic Science and the Crafoord Prize winner Maynard Smith in biology game theory [92].

Game theory has different forms and properties depending on its applications. In this section some important properties are introduced, including cooperative and non-cooperative game, zero-sum and non-zero-sum game and simultaneous and sequential gaming.

Cooperative game represents a game with players from alliances, while non-cooperative game refers to a game with competitive and self-enforcing players [92]. Cooperative game focuses on predicting joint actions and the resultant payoffs, while non-cooperative game focuses on predicting individuals' actions, payoffs and analysing the Nash equilibrium [11, 15]. The non-cooperative game is usually more general and operates under sufficient assumptions that a possible strategy can be simulated. However, in many cases the resulting model has high complexity as a practical tool in the real world. To overcome this issue, cooperative game provides simplified approaches that cut down the number of assumptions and reduce the complexity of the problem.

Zero-sum game represents the scenario that the total benefit of all players sums up to zero [Owen]. Most board games are actually zero-sum games, including Go and chess, where the number of a single player's wins is exactly the number of the other player's losses. In non-zero-sum games, the gain of one player is not necessarily the loss of the other player. Non-zero-sum game can be applied in many cases such as in an economic situation and the well-known prisoner's dilemma.

Simultaneous game is when players move together while sequential game is when players take turns. The key difference between these two games is whether or not players are aware of others' actions at the same or close time step. In a simultaneous game, a player makes his or

her decision without considering others' current actions and vice versa. The extension of such a concept can be discussed using the notion of perfect information and imperfect information. The perfect information game is when all players know all the moves of the other players. Such a condition is difficult to achieve in simultaneous games, as decisions are usually made at the same or close to the same time. Sequential game can also be an imperfect information game such as poker or contract bridge in card games [99]. Complete information is sometimes compared with perfect information, where the player knows the available strategies and payoffs to other players but not the actions [119]. Games with multiple moves are called combinatorial games. Games with imperfect information usually have combinatorial characteristics. This type of model has been addressed in many studies on artificial intelligence [10, 60].

In this work, the property of games will be designed following the completion of behaviour models. Consideration of different properties will be compared with reasoning to match behaviours.

The study of economics and business has resulted in considerable interest in mathematical modelling through game theory [118], such as auctions, bargaining mergers acquisitions pricing [3], behavioural economics [37] and information economics [106]. Most studies focus on finding a solution to a set of strategies under the assumption that players act rationally. In non-cooperative games, such a solution is called the Nash equilibrium, where all players give their best responses to other players. In such cases, players do their best to utilise their position and resources to understand different strategies from others [26, 25]. Common procedures of models are built following three steps: 1) Presenting a game of a situation, 2) One or more solutions are shown, 3) Demonstration of equilibrium.

Related studies such as decision theory and operations research have similar disciplines using Markov Decision Processes (MDP) [103]. These studies are usually referred as one-player games, which can be included within game theory by adding randomly acting players [97]. Games with the property of MDP can be understood through observing stochastic outcomes, an important area of research in AI.

In this work, the design of game theory model will be based on the output of a behaviour simulation, shown in Figure 6.4. The behaviour simulation and game theory modelling will be combined as a simulation of a trade strategy between participants. In the case of the food market, the participants are consumers retailer and manufacturer.

After designing the game simulation with consumer behaviour, I aim to apply an AI agent to capture optimal decisions. An approach called reinforcement learning is introduced in the next section.

With the inspiration of behaviour psychology, reinforcement learning is introduced in the field of machine learning with the objective of finding the optimal actions that maximise

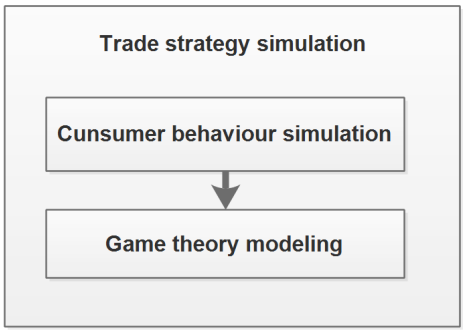


Fig. 6.4 Trade strategy simulation architecture, containing consumer behaviour simulation and game theory modelling.

cumulative reward in certain environment. Markov decision process (MDP) is typically used to describe an environment that machines are exposed [132]. A reinforcement learning algorithm does not require full knowledge of MDPs while other techniques take far more time to understand them. Instead of training with correct fixed input/output datasets, reinforcement learning uses interactions between decisions and rewards to train the model. It focuses on on-line performance and balancing between exploring unknown decisions and exploiting current knowledge of MDPs [62].

Considering a basic model of reinforcement learning, I start by defining MDPs. First, I define $\mathbf{s}$ as the states of the agent in an environment, and $\mathbf{A}$ as the action set of the agent. I define the probability of transition from state $\mathbf{s}$ to $\mathbf{s}'$ under action $\mathbf{a}$ using Equation 6.1 .

$$P_a(s,s') = Pr(s_{t+1} = s' | s_t = s, a_t = a) \tag{6.1}$$

Furthermore, I express immediate reward after transition from $\mathbf{s}$ to $\mathbf{s}'$ with action $\mathbf{a}$ as $\mathbf{R_a}(\mathbf{s},\mathbf{s}')$. Finally, I design the rules and functions of the observable variables to my model based on needs and applications. The rules and functions are stochastic and the observation is usually the immediate rewards of the last transition.

An agent of reinforcement learning interacts with the environment in discrete time steps. The agent receives observation $\mathbf{o_t}$ at each time step $\mathbf{t}$, which is expected to include reward $\mathbf{r_t}$. From the available movement, an action $\mathbf{r_t}$ is selected and sent to the environment. After the environment receives the action, it moves to a new state $\mathbf{s_{t+1}}$ and generates a reward $\mathbf{r_{t+1}}$ with the transition $(\mathbf{s_t}, \mathbf{a_t}, \mathbf{s_{t+1}})$. A reinforcement learning agent aims to maximise the collection of rewards, where the available actions are derived from history or randomised action selection. Regret is generated during comparison of two moves or between the performance of two agents. With the goal of maximising the collected rewards, the agent must consider long term consequences even though the immediate reward is negative. Two procedures are followed

by the algorithm, which optimises performance with collected samples and approximates its function towards the environments.

With the property described above, reinforcement learning is frequently used to solve problems, with a trade-off made between long-term and short-term rewards [125].

Policy π is a map of probability of taking action in state $\mathbf{s}$ [125]. This map is used to model an agent's action selection, as shown in Equation 6.2.

$$\pi(a|s) = P(a_t = a|s_t = s) \tag{6.2}$$

Value function is expressed as $\mathbf{V}_\pi(\mathbf{s})$ with state $\mathbf{s}$ and policy π, and is defined by expected return $\mathbf{E}[\mathbf{R}]$, as shown in Equation 6.3.

$$V_\pi(s) = E[R] \tag{6.3}$$

$\mathbf{R}$ is a random variable for return, as expressed in Equation 6.4, where $\gamma^{\mathbf{t}}$ is the discount rate at time $\mathbf{t}$ and $\mathbf{r_t}$ is the reward.

$$R = \sum_{t=0}^{\infty} \gamma^t r_t \tag{6.4}$$

The aim of the designed algorithm is to maximise the expected return. I express value function formally as Equation 6.5, where $\mathbf{R}$ is expressed as the random return following policy π from the initial state $\mathbf{s}$.

$$V^\pi(s) = E[R|s,\pi] \tag{6.5}$$

The optimal value is defined in Equation 6.6, where all policies that achieve $\mathbf{V}^*(\mathbf{s})$ are called optimal solutions.

$$V^*(s) = \max_\pi V^\pi(s) \tag{6.6}$$

It is expected that the optimal policy also maximises the expected return ρ^π from the randomly sampled state $\mathbf{S}$.

Finally, the action value is defined as $\mathbf{Q}^\pi(\mathbf{s},\mathbf{a})$ given a policy π, a state $\mathbf{s}$ and an action $\mathbf{a}$, as shown in Equation 6.7, where $\mathbf{R}$ is the random return following the policy, state and action.

$$Q^\pi(s,a) = E[R|s,a,\pi] \tag{6.7}$$

In the theory of MDP, an optimal policy π^* can lead to an optimal action by choosing from the list of $\mathbf{Q}^{\pi^*}(\mathbf{s},\cdot)$, which can be denoted as $\mathbf{Q}^*$.

Under the assumption of full knowledge of the MDP, the optimal action value function can be derived by value iteration and policy iteration [125].

Before solving the MDP, I extend Equations 6.2 and 6.3 to Equations 6.8 and 6.9 to describe the two steps required to reach optimal solutions.

$$\pi(s) := argmax_a \left\{ \sum_{s'} P_a(s,s')(R_a(s,s') + \gamma V(s')) \right\} \tag{6.8}$$

$$V(s) := \sum_{s'} P_{\pi(s)}(s,s')(R_{\pi(s)}(s,s') + \gamma V(s')) \tag{6.9}$$

In value iteration, π function is not applied and is assumed to be a calculated value, which is used to substitute Equation 6.9. A combined function is generated after the substitution, as shown in Equation 6.10.

$$V_{i+1}(s) := \sum_{s'} P_a(s,s')(R_a(s,s') + \gamma V_i(s')) \tag{6.10}$$

where i represents the iteration number starting from 0. The repetition of computing $\mathbf{V_{i+1}}$ for all states $\mathbf{s}$ brings $\mathbf{V}$ to convergence. In such a case Equation 6.10 is called the Bellman Equation [8].

In policy iteration, there are two steps: policy evaluation and policy improvement, where Equation 6.8 is applied once and Equation 6.9 is repeated. When applying methods in reinforcement learning, the action value representation of $\mathbf{Q}$ is used as in Equation 6.11 [125].

$$Q(s,a) = \sum_{s'} P_a(s,s')(R_a(s,s') + \gamma V_i(s')) \tag{6.11}$$

The Monte Carlo method is frequently used in the policy evaluation step. All function values $\mathbf{Q}^{\pi}(\mathbf{s},\mathbf{a})$ are computed under a given policy π for all state-action sets $(\mathbf{s},\mathbf{a})$. Such computation assumes that the MDP is finite and that sufficient memory is available for all episodic steps from some random initial state and steps extended from the state. The precise estimates $\mathbf{Q}$ of action-value function $\mathbf{Q}^{\pi}$ are expected to be generated under sufficient time. In the step of policy improvement, the next policy is obtained through a greedy algorithm with respect to $\mathbf{Q}$, where the new policy returns an action that maximises $\mathbf{Q}^{\pi^*}(\mathbf{s},\cdot)$ given a state $\mathbf{s}$.

Some common issues might occur during the procedure of using the Monte Carlo method, such as time wasted evaluating a sub-optimal policy, inefficient sampling on a long trajectory with single state-action and difficulty for convergence in trajectories with high variance. In order to overcome these issues, the Temporal Difference method is introduced. This method enables the variation of the policy before the algorithm converge which avoids wasting time on

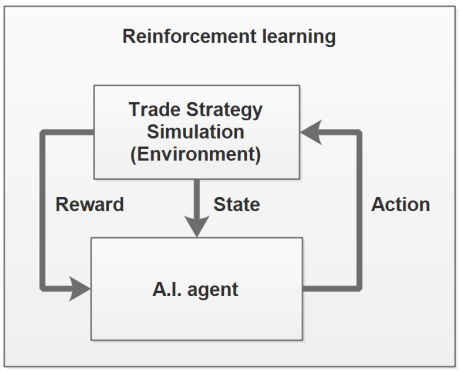


Fig. 6.5 Reinforcement learning AI agent observes a state from a trade strategy simulation, performs an action and receives rewards.

sub-optimal policies. Another adjustment of this method is the contribution of trajectories to all state-actions on the same path, which can solve inefficient sampling with single state-action and the difficulty of convergence on a trajectory [125]. However, the Temporal Difference method may lose generality and efficiency when returns are noisy and it relies on the Bellman equation for value iteration. Therefore, a λ parameter is sometime introduced, where the value is between 1 and 0, as a trade-off of using Monte-Carlo and the Temporal difference method.

Furthermore, the Monte Carlo method can only be applied in episodic and finite environments [125]. In order to address these limitations, function approximation methods have been proposed. In linear approximation form, the action value function is expressed in Equation 6.12, where ϕ is a finite set of vectors for a state-action pair multiplied by some weights θ. The algorithm adjusts the weights instead of adjusting the value of the individual state-action pairs.

$$Q(s,a) = \sum_{i=1}^{d} \theta_i \phi_i(s,a) \tag{6.12}$$

Figure 6.5 shows the approaches and methods that have been considered for in this work. The trade strategy simulation from previous sections is described as the MDP and treated as the environment for my designed AI agent. Using a reinforcement learning algorithm, the AI agent receives rewards and states based on the series of actions it takes in the environment. It is expected to discover optimal solutions based on observing the trade strategy simulation.

With the realistic simulation of a behavioural game environment, it is possible to study consumers', retailers' and manufacturers' interactions as shown in Figure 6.6. First I study the generated simulations, design a model to capture factors, and implement an AI agent to generate recommendations for all different layers. Finally, I apply my previous research "Personalised Expert Recommendation System on Optimised Nutrition" to the simulation to study the influence of the environment. In previous research, I have demonstrated that healthier

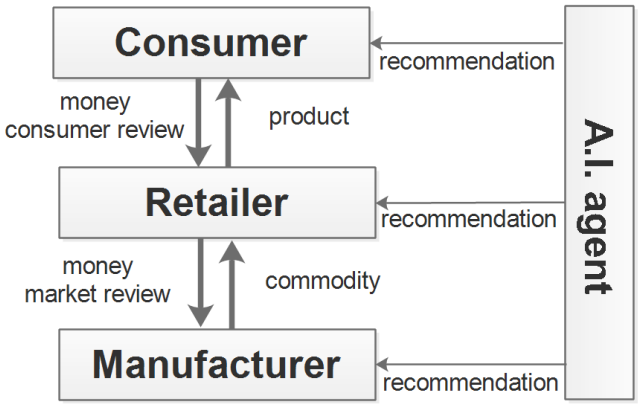


Fig. 6.6 A simplified framework of the Grocery Business Unionism Strategy in Neural Expert System Simulation (GBUSiNESS). This system simulation provides recommendations not only to consumers but also to retailers and manufacturers.

food can be chosen by recommending personalised product selections. In the next stage, I aim to generate recommendations that influence retailers and manufacturers towards better food making and sales. This research can potentially be extended to the concept of Nudgeomic and provide evidence of its powerful influence in the food industry. It is expected that this research will benefit not just consumers but also retailers and manufacturers.

In order to define the economic phenomena that are relevant in a supermarket simulation, the concept of business unionism is included in this work. A business union describes a trade union that seeks for immediate economic interests. Business unionism aims to study the effects and influence of a business union [53]. The concept was originally used to describe the behaviour of labourers in a company, similar to my work on the behaviour of manufacturers in a supermarket: each manufacturer (an employee/labourer) receives a profit from the supermarket (the employer/company). The manufacturers can compete with each other to achieve more sales, or work together to achieve higher prices for all of the available products in the market.

Each manufacturer agent seeks a constant and immediate increase in profit to improve his or her living conditions. This action is straight-forward and short-term, as described in the explanation of business unionism [108]. In business unionism, the action usually results in a process of intensive and collective bargaining under rigid specifications. For a manufacturer-supermarket scenario, this corresponds to the trade-off process between manufacturers and consumers, under the defined rules of the grocery product purchase game. This concept inspired me to work on a framework that simulates interactions between consumers and manufacturers, using the concept of business unionism, and to study the influence that may be brought to bear on consumers, manufacturers and supermarkets.

Fig. 6.7 Triple constraints for consumer satisfaction when selecting a product to purchase. This design of the consumer's mindset trades-off the factors of health, taste and price.

6.2 Simulation Architecture

Following the architecture described in the previous section, I create simulated customers based on assumptions and an adept reinforcement learning AI agent to learn and simulate customer behaviours in real life. To demonstrate the concept, I experiment with a simple scenario: a world with few consumers and few food product manufacturers. For each time step, consumers are free to choose a product that is provided by the small number of manufactures, and each manufacturer can only make one product, as shown in Figure 6.8.

The simulation continues to loop through four steps: 1) Customers observe product prices and their nutrition values. 2) Customers make a decision to choose one of the three products to purchase. 3) Once all four customers have made their decision, the manufacturers observe the sales of their products and derive profit. 4) Manufacturers modify their prices and nutrition values by considering sales and profits.

I then focus on individual components within the scenario. Based on the general reinforcement learning architecture in Figure 6.5, a consumer behaviour model is developed to fit into the AI agent component. The action of each consumer (i.e. AI agent) is to choose a product. The state of each consumer is governed by the prices and nutrition values of available products. Finally, the reward of each consumer is defined by the benefit of the price difference between other products or the superior nutrition combination. This is shown in Figure 6.9.

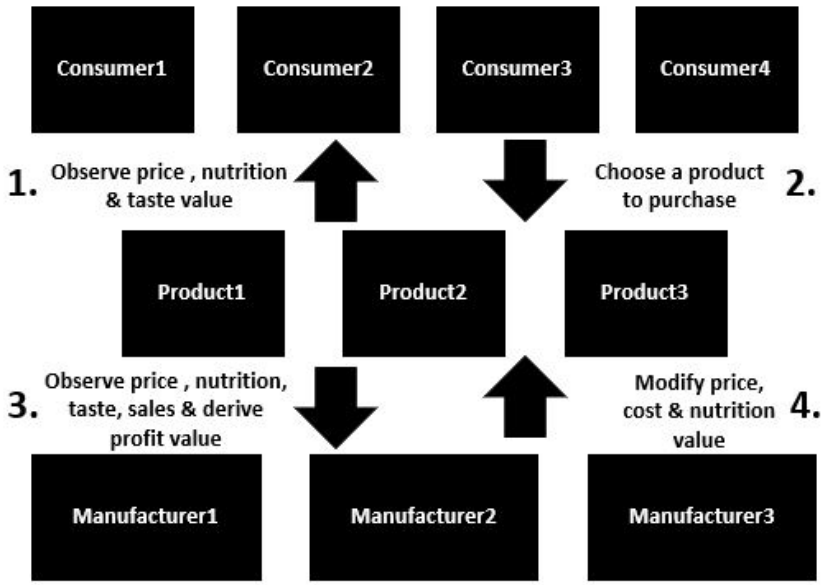


Fig. 6.8 The design of interaction between consumers, products and manufacturers involves few actions. The consumer observes product features and chooses a product to purchase. The manufacturer observes product sales, derives profit and modifies product features.

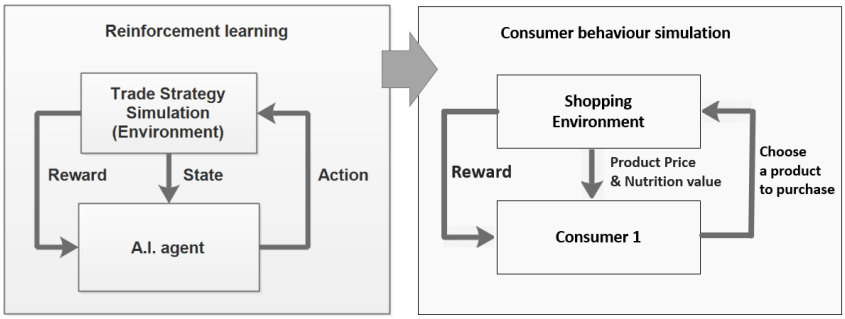


Fig. 6.9 The consumer behaviour simulation agent model is trained following regular reinforcement learning techniques. The state of a consumer is defined by the available product features and the action is equivalent to choosing a product to purchase. The reward is derived from the product features and action that the agent performs.

I initiate the consumer model by looking into the products, which is the input to the consumer models. For each product, a product matrix $\mathbf{P}$ is created, which can be expressed as a vector set $\mathbf{p}_i$ or $\mathbf{q}_f$, see Equation 6.13. For $\mathbf{p}_i$, the value p_f is the feature value where f is the label of the feature, which can be further expressed with the names of the features of interest. In the case of having three products, the product feature is defined as $PF = [price, health, taste]$ and $\forall f \in PF$. The vector set $\mathbf{q}_f$ contains a series of elements q_i where i is the product identification number, representing different products. In my case, the product list is expressed

as $PL = [1,2,3]$ and $\forall i \in PL$. Note that all $\mathbf{q}_f$ are standardised into Z-score format. $\mathbf{p}_i$ and $\mathbf{q}_f$ are the same matrix, aiming to express different directions for the ease of subsequent row or column operations. For example, looking at $\mathbf{p}_i$ I can express different products with label i while the product feature f is hidden. On the other hand, the $\mathbf{q}_f$ can express the product feature f and hide the product label i.

$$\begin{aligned}\mathbf{P} = \mathbf{p}_i = [p_1, p_2, ..., p_f, ..., p_F]_i = [p_{price}, p_{health}, p_{taste}]_i \\ = \mathbf{q}_f = [q_1, q_2, , ..., q_i, ..., q_I]_f = [q_1, q_2, q_3]_f\end{aligned} \tag{6.13}$$

Inspired by the field of nutrigenetics and my proposed framework PERSON as discussed in Section 2.2 and in section 4.4, I consider each nutrient as a separate feature factor when generating personalised health recommendations. Therefore the health factor feature is described by different approaches in Equations 6.14 and 6.15. In Equation 6.14 with the personalised approach, I consider the health feature as a list of nutrients with their individual values. In Equation 6.15 with the generalised approach, I consider the health feature as one value derived from the mean of all nutrients, minus a bias value.

$$p^{personalised}_{Health,i} = [p_{Energy,i} - Bias_i, p_{Fat,i} - Bias_i, ...] \tag{6.14}$$

$$p^{generalised}_{Health,i} = mean(p_{Energy,i} - Bias_i, p_{Fat,i} - Bias_i, ...) \tag{6.15}$$

Considering Equations 6.13 and 6.14, the product feature matrix with a personalised approach can be rewritten as Equation 6.16. Moreover, considering Equations 6.13 and 6.15, the product feature matrix with a generalised approach can be rewritten as Equation 6.17.

$$\mathbf{p}^{personalised}_i = [p_{Price}, p_{Taste}, p^{personalised}_{Health}]_i = [p_{Price}, p_{Taste}, p_{Energy} - Bias, p_{Fat} - Bias, ...]_i \tag{6.16}$$

$$\mathbf{p}^{generalised}_i = [p_{Price}, p_{Taste}, p^{generalised}_{Health}]_i = [p_{Price}, p_{Taste}, mean(p_{fat} - Bias, p_{sugar} - Bias, ...)]_i \tag{6.17}$$

$$p_{taste} = Bias_i + \frac{1}{5g_n}(e^{|2.625 s_{Energy,i}|} + e^{|2.625 s_{Fat,i}|} + ... + e^{|2.625 s_{n,i}|} + ...) \tag{6.18}$$

After replacing the health feature with specific nutrition features, the product feature PF is extended to $PF = [Price, Taste, Energy, Fat, Sugar, Salt, Carbohydrate]$. In order to express how the taste would influence a consumer's decision in my designed simulation, a taste

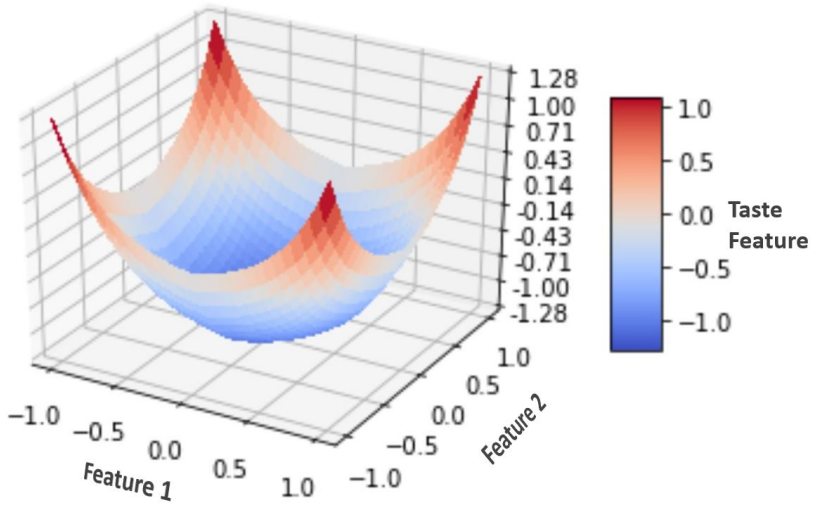


Fig. 6.10 The relationship between taste and a variety of nutrition features. This three-dimensional plot shows an example of two features influencing taste.

feature is introduced as p_{taste}, which has a relationship with the nutrition related features, expressed as in Equation 6.18, where $\forall n \in NF$ and NF is the nutrition feature defined as $NF = [Energy, Fat, sugar, salt, Carbohydrate]$. In Equation 6.19, $s_{n,i}$ is defined as the percentage of modification of a feature. For example, $s_{sugar,i}$ is the percentage of sugar modified from its original amount. g_n is the number of features that are involved with taste. Bias is an offset parameter that has a relationship with the cost of making the product. Assuming that bias is fixed at 1.48, looking at two nutrition parameters: sugar and fat, and ignoring other nutrition factors, I can plot the taste parameter in relation to the variation of the two nutrition parameters, while fixing all other parameters to their original values, as shown in Figure 6.10.

$$s_{n,i} = \frac{p_{n,i}^{original} - p_{n,i}^{new}}{p_{n,i}^{original}} \tag{6.19}$$

Equation 6.18 is designed based on assumption and trial-and-error, which can be replaced in subsequent work if there is data available to model this relationship.

Figure 6.10 shows that the taste feature is influenced by the variation of nutrition features. When both features are unmodified, the taste is at its maximum value and any modification will decrease the taste. This is under the assumption that the manufacturer's original product is at its best taste condition before any modification. The bias in Equation 6.18 is a term that offsets the score of the taste feature, which is expressed as in Equation 6.20. I assumed that the taste feature is also influenced by the bias that manufacturers give to consumers. This bias

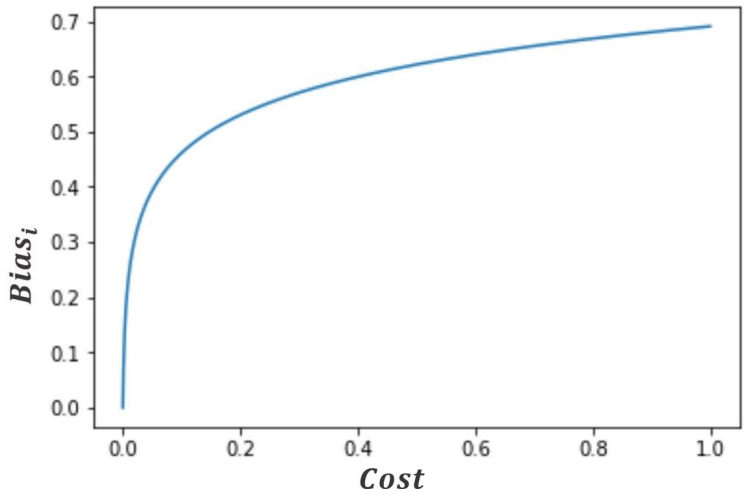


Fig. 6.11 The relationship between bias and cost. Both values are normalised and can vary from 0 to 1.

has a natural log relationship to cost, as shown in Figure 6.11. All values in this equation are normalised.

$$Bias_i = \frac{ln(Cost)}{10} \tag{6.20}$$

Following the triple constrain model in Figure 6.7, I have described the relationships between health, price and taste. The next step is to experiment with the consumer behaviour model, while considering consumers' actions and rewards and to establish benefit boundaries.

The consumer model observes the product feature vector and outputs an action vector $\mathbf{a}_c$, where c represents different consumers, as expressed in Equation 6.21. The consumer list is expressed as CL and $\forall c \in CL = [1,2,3,4]$, and i represents the available products.

$$\mathbf{a}_c = [a_1, a_2, ..., a_i, ..., a_I]_c = [a_1, a_2, a_3]_c \tag{6.21}$$

Once the shopping environment model receives action vectors from consumers, it feeds back a reward to each consumer, which is presented as vector $\mathbf{R}$, see Equation 6.22. In the equation, c varies from 1 to 3, representing three consumers as an example.

$$\mathbf{R} = [r_1, r_2, ..., r_c, ...r_C] = [r_1, r_2, r_3] \tag{6.22}$$

The value of the reward is generated from a function defining the benefits that a consumer will receive after choosing a product. In order to derive this equation, I introduce a matrix of a benefit boundary for each consumer $\mathbf{b}_c$, see Equation 6.23.

$$\mathbf{b}_c = [b_1, b_2, ..., b_n, ..., b_N]_c = [b_{price}, b_{health}, b_{taste}]_c \tag{6.23}$$

Considering the personalised and generalised recommendations, the benefit boundary matrix can be rewritten as Equations 6.24 and 6.25.

$$\mathbf{b}_c^{personalised} = [b_{Price}^l, b_{Taste}^l, b_{Energy}^l, b_{Fat}^l, ..., b_f^l, ...]_c \tag{6.24}$$

$$\mathbf{b}_c^{generalised} = [b_{Price}^l, b_{Taste}^l, b_{health}^{generalised,l}]_c \tag{6.25}$$

Inspired by Table 4.4, I introduce a level list L consisting of three levels of benefits based on the consumer's preference: $\forall l \in L = [High(H), Medium(M), Low(L)]$. The boundary matrices of high, medium and low are set using Equations 6.26, 6.27 and 6.28. Considering Equation 6.13, the product matrix is expressed in the form of $\mathbf{q}_f$ where f is the product feature and $\mathbf{q}_f$ is the vector of a feature of all products.

$$b_f^H = Mean(\mathbf{q}_f) + Std(\mathbf{q}_f) \tag{6.26}$$

$$b_f^M = Mean(\mathbf{q}_f) \tag{6.27}$$

$$b_f^L = Mean(\mathbf{q}_f) - Std(\mathbf{q}_f) \tag{6.28}$$

The above definitions cover most components for the boundary of the matrix. Considering the personalised and generalised advice, the personalised boundaries can be expressed as Equation 6.29, 6.30 and 6.31. Where $\forall n \in NF$ and NF is the nutrition feature, defined as $NF = [Energy, Fat, Sugar, Salt, Carbohydrate]$

$$b_n^{personalised,H} = Mean(\mathbf{q}_n^{personalised}) + Std(\mathbf{q}_n^{personalised}) \tag{6.29}$$

$$b_n^{personalised,M} = Mean(\mathbf{q}_n^{personalised}) \tag{6.30}$$

$$b_n^{personalised,L} = Mean(\mathbf{q}_n^{personalised}) - Std(\mathbf{q}_n^{personalised}) \tag{6.31}$$

extending from Equation 6.14, I can express $\mathbf{q}_n^{personalised}$ as in Equation 6.32.

$$\mathbf{q}_n^{personalised} = [p_{n,1} - Bias_1, p_{n,2} - Bias_2, ...p_{n,i} - Bias_i, ...p_{n,I} - Bias_I] \tag{6.32}$$

On the other hand, $b_{health}^{generalised,l}$ is obtained differently because it considers all nutrition values as one health feature factor, following Equation 6.15.

$$b_{health}^{generalised,H} = Mean(\mathbf{q}_{Health}^{generalised}) + Std(\mathbf{q}_{Health}^{generalised}) \tag{6.33}$$

$$b_{health}^{generalised,M} = Mean(\mathbf{q}_{Health}^{generalised}) \tag{6.34}$$

$$b_{health}^{generalised,L} = Mean(\mathbf{q}_{Health}^{generalised}) - Std(\mathbf{q}_{Health}^{generalised}) \tag{6.35}$$

From Equation 6.15, I can express $\mathbf{q}_{Health}^{generalised}$ as in Equation 6.36.

$$\mathbf{q}_{Health}^{generalised} = [p_{health,1}^{generalised} - Bias_1, p_{health,2}^{generalised} - Bias_2, ...p_{health,i}^{generalised} - Bias_i, ...p_{health,I}^{generalised} - Bias_I] \tag{6.36}$$

The values of these boundaries are inspired by the expert system introduced in Section 4.4 and are directly established by a derivation of the statistics of product data. The generation of this matrix is explained in Chapter 4.3. In Equation 6.37, I introduce a filtering function, in which I look at the difference $d_{c,f}$ of the product feature of the chosen product $p_{f,chosen_c}$, and the benefit boundary $b_{f,c}$.

$$d_{c,f} = p_{f,chosen_c} - b_{f,c} \tag{6.37}$$

In Equation 6.38, I define the reward r_c, for consumer c with a series of summations of $d_{c,f}$. If any of the nutrition, taste or price features exceed their benefit boundaries, the variable $d_{c,f}$ will output a negative value resulting in a decrease of the reward. The weights of features W_f are set to values that emphasise the contrasting importance of different features.

$$r_c = \sum_{f=1}^{\mathbf{N}} W_f d_{c,f} \tag{6.38}$$

The $\mathbf{p}_{f,chosen_c}$ is derived from the action $\mathbf{a}_c$ and product feature $\mathbf{q}_f$, see Equation 6.39. In the action vector set $\mathbf{a}_c$, only one element can be 1 and the rest of the element is 0, demonstrating that the consumer can only make one decision at a time.

$$p_{f,chosen_c} = \mathbf{q}_f \odot \mathbf{a}_c^T = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}_f \odot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}_c \tag{6.39}$$

After defining the input and output setup for the consumer model, the Deep Q-learning (DQN) is implemented for the model, which is a deep neural network that mimics the effects of reactions receptive fields [90]. The Deep Neural Network (DNN) architecture was introduced in Section 4.2. The same architecture is applied here together with a memory storing current state, future state, action and reward sets, which can be expressed as Equation 6.40. The $\mathbf{s}_c$ is the current state which is what the consumer observes regarding the products features $\mathbf{p}_i$ at this time step, as expressed in Equation 6.13. The $\mathbf{a}_c$ is the action carried out by consumer c and the $\mathbf{r}_c$ is the reward that the consumer c receives after executing action $\mathbf{a}_c$. The $\mathbf{s'}_c$ is the next state, which is the environment's response after observing the action from the consumer.

$$\mathbf{z}_c = [\mathbf{s}_c, \mathbf{a}_c, \mathbf{r}_c, \mathbf{s'}_c] = [\mathbf{p}_i, \mathbf{a}_c, \mathbf{r}_c, \mathbf{p'}_i] \tag{6.40}$$

Figure 6.12 shows a single iteration of the DNN model. For every iteration, two main steps occur: **Step one:** the DNN observes the current state $\mathbf{s}_c$ and performs an action $\mathbf{a}_c$, the Memory component not only takes note of current state $\mathbf{s}_c$ and action $\mathbf{a}_c$, but also receives reward $\mathbf{r}_c$ and the next state $\mathbf{s'}_c$ from the environment. **Step two:** the DNN is switched to a training mode and is trained by the memory, aiming to find the $Q^{\pi^*}(\mathbf{s}, \mathbf{a})$ that satisfies the Bellman equation, see Equation 6.41.

$$Q^{\pi^*}(\mathbf{s}, \mathbf{a}) = r + \gamma \max_{\mathbf{a'}} Q(\mathbf{s'}, \mathbf{a'}) \tag{6.41}$$

The optimisation goal is to minimise the mean squared error of $Q^{\pi^*}(\mathbf{s}, \mathbf{a})$ and $r + \gamma \max_{\mathbf{a'}} Q(\mathbf{s'}, \mathbf{a'})$, which is expressed as L, as shown in Equation 6.42.

$$L = \frac{1}{2}[r + \gamma \max_{\mathbf{a'}} Q(\mathbf{s'}, \mathbf{a'}) - Q^{\pi^*}(\mathbf{s}, \mathbf{a})] \tag{6.42}$$

The function Q is a representation of the DNN to minimise L. In order to implement DNN to execute this optimisation task, a DNN is built for each consumer agent, see Figure 6.13. The product information is sent to a prepossessing layer before being fed to the DNN. Assuming that consumers in the real world observe the relative values of nutrition, price and taste, the mean value of all available products is calculated and each product's feature values to the feature's means are derived, see Equation 6.43. In other words, the model is under the condition that one must choose one product and that this wukk be the best for oneself. The input health feature in Figure 6.13 is a single feature under the condition that consumers consider general health decisions. It splits into multiple nutrition features under the condition that consumers consider personalised decisions.

$$Input_{f,i} = Mean(\mathbf{q}_f) - p_{f,i} \tag{6.43}$$

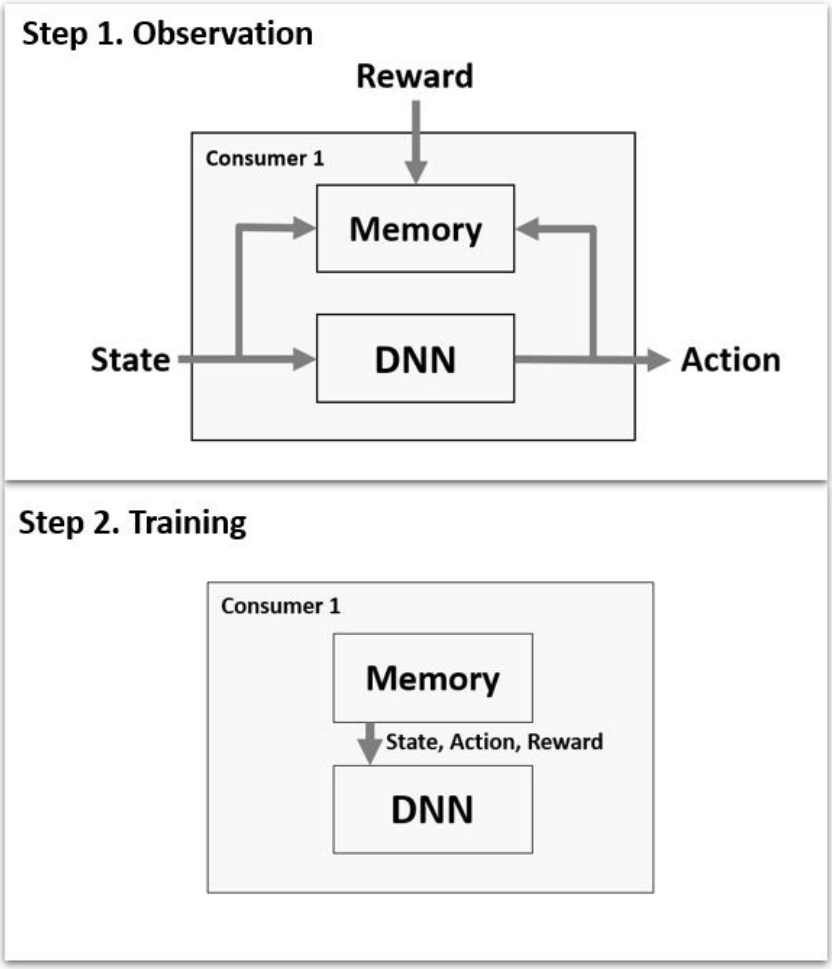


Fig. 6.12 Two different states of the designed AI agent. In the observation state, the agent performs an action based on state observation and receives a reward. The state, action and reward are stored in the memory component. In the training state, the AI agent updates its deep neural network by iterating data from the memory component.

For each neuron, each layer is defined by Equation 6.44, where j is the number of layers, in is the input label and out is the output label, NW is the neural network weight matrix.

$$\mathbf{h}_{out}^{j} = f_{relu}(\mathbf{NW}_{out,in}\mathbf{h}_{in}^{j}) \tag{6.44}$$

Assuming a customer agent consists of three layers in the DNN, by plugging the neuron to the input feature and output action, we can obtain $\mathbf{h}_{in}^{1,c} = Input_{f,i}$ and $\mathbf{h}_{out}^{3,c} = \mathbf{a}_c$.

In the training process, the output action vector is compared with a label $\mathbf{y}$, which is equal to an expected action $a_c^{expected}$, see Equation 6.45.

$$\mathbf{y}_c = \mathbf{a}_c^{expected} \tag{6.45}$$

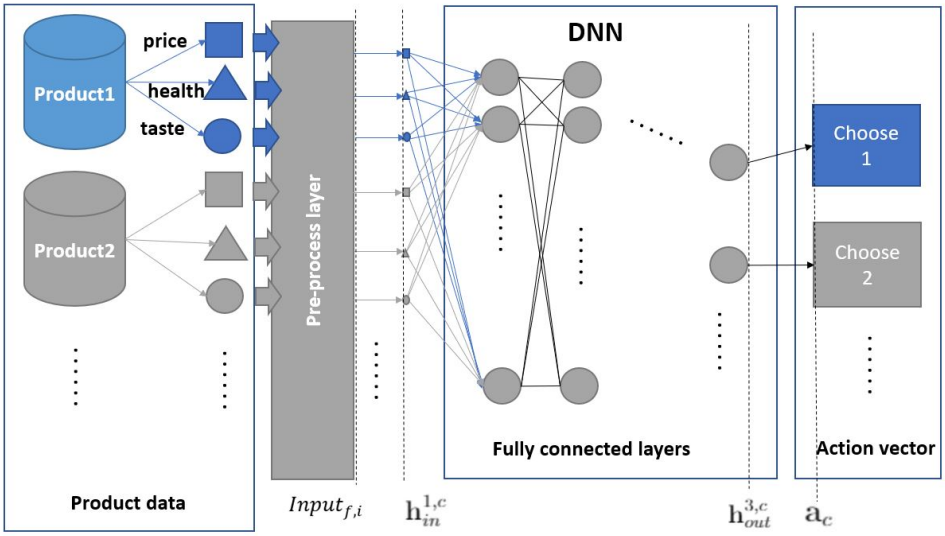


Fig. 6.13 The deep neural network structure of a consumer AI agent observes all product information from the left side of the figure through a pre-processing layer and makes a decision on purchasing one product through generating an action vector.

The expected action is derived from the reward and the response from the environment, see Equation 6.46. The expected action is the action of the neuron plus a target reward vector $\mathbf{r}_c^v$.

$$\mathbf{a}_c^{expected} = \mathbf{a}_c + \mathbf{r}_c^v = [a_1, a_2, ..., a_i, ..., a_I]_c + [0, 0, r_g, ..., 0]_c \tag{6.46}$$

$$\mathbf{r}_c^v = [0, 0, r_g, ..., 0]_c = r_c \mathbf{u}_g \tag{6.47}$$

$\mathbf{u}_g$ is a directional unit vector that points in g direction, where $g = \mathbf{argmax}(\mathbf{a}_c)$, is the location of the action that the agent decides to perform.

In conclusion, the optimisation process of a target is as follows: if the action is good, I feed the reward to the target action. If the action is bad, the reward turns to a negative value and a punishment for the action.

To extend the optimisation to cover future states, the future state **s'** is used to determine future action $\mathbf{a'}_c$ with the same neurons. Another variable called the confidence factor $\mathbf{f}_c = \mathbf{max}(\mathbf{a'}_c)\mathbf{u}_g$ is introduced to express the confidence of the action in the future state **s'**. Therefore Equation 6.46 is re-written as Equation 6.48 and γ is a parameter used to determine the learning rate involved with the future states.

$$\mathbf{a}_c^{expected} = \mathbf{a}_c + \mathbf{r}_c^v + \gamma \mathbf{f}_c \tag{6.48}$$

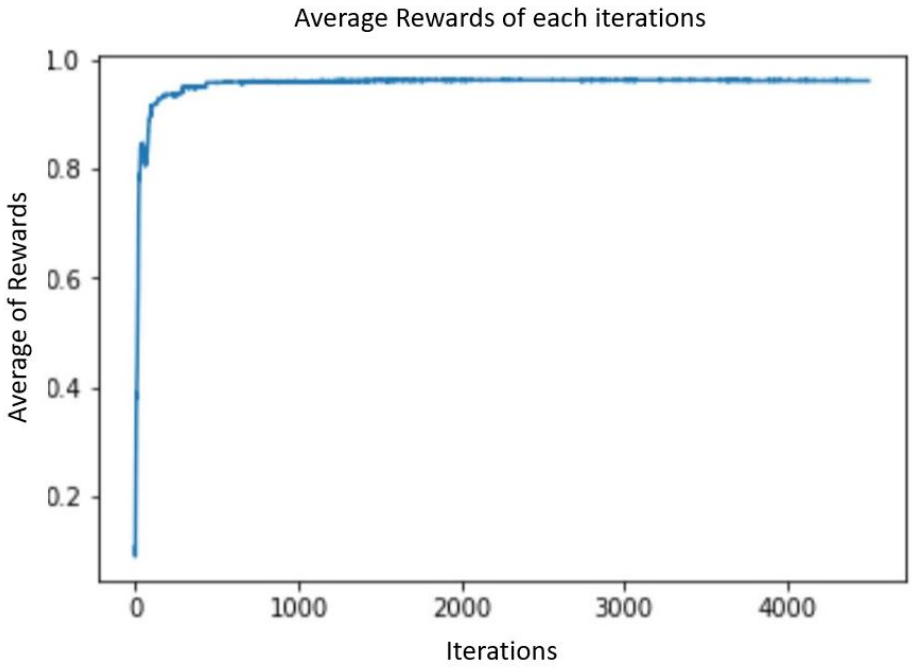


Fig. 6.14 The average reward of a consumer agent facing a random state and performing its action. The horizontal axis represents the training iteration. The average reward reaches its maximum value when the optimum solution is performed by the consumer agent.

As shown in Figure 6.12, the memory component $\mathbf{z}_c$ memorises multiple sets of $[\mathbf{s}_c, \mathbf{a}_c, \mathbf{r}_c, \mathbf{s'}_c]$ and loops through these historical records to train the consumer agent. The Stochastic Gradient Descent (SGD) method is applied as the optimiser to update the model with $\mathbf{y}_c$ and $\mathbf{a}_c$. In other words, to consider the difference between $\mathbf{a}_c^{expected}$ and $\mathbf{a}_c$.

Training is undertaken through iterations of feeding and rewarding. The training sets are randomly generated product combinations, the features of which vary between 50% and 150% of the average values of existing products. The testing set contains 1,000 randomly generated products.

For each iteration, the trained model is ready to give answers to the test set and to record the average rewards. The average of collected rewards versus iterations is shown in Figure 6.14. The average reward reaches its maximum value of 0.94 after 2,125 iterations, and becomes stable at 4,232 iterations.

The boundaries and the weights of the features are the main sources of differentiation between the characteristics of the consumer agents. The boundary influences how the agent is trained, while the weight represents the feature's importance. For demonstration purposes, a simplified experiment of three different consumers with different boundaries $\mathbf{b}_{f,l}$, and three different products with different features is introduced. This experiment observes how consumers

Features in z-score x=1 / Product features \ Benefit boundary of Consumer	W_{all}=1 $b_{sugar,high}$ =0.463 $b_{taste,high}$ =0.5 $b_{other,high}$ =0.5	W_{all}=1 $b_{sugar,medium}$ =0.3 $b_{taste,medium}$ =0.5 $b_{other,medium}$ =0.5	W_{all}=1 $b_{sugar,low}$ =0.1367 $b_{taste,low}$ =0.5 $b_{other,low}$ =0.5
Product1: $p_{sugar,1}$ = 0.5 $p_{taste,1}$ = 0.5 $p_{other\ features,1}$ = 0.5	$r_{1,high}$= -0.0367	$r_{1,medium}$= -0.2	$r_{1,low}$= -0.363
Product2: $p_{sugar,2}$ = 0.3 $p_{taste,2}$ = 0.5 $p_{other\ features,2}$ = 0.5	$r_{2,high}$= 0.163	$r_{2,medium}$= 0	$r_{2,low}$= -0.163
Product3: $p_{sugar,2}$ = 0.1 $p_{taste,2}$ = 0.5 $p_{other\ features,2}$ = 0.5	$r_{3,high}$= 0.363	$r_{3,medium}$ = 0.2	$r_{3,low}$ = 0.0367

Fig. 6.15 The designed consumer experiment with different nutritional boundaries. The vertical axis presents various products setup and the horizontal axis presents the different benefit boundaries of consumers. The rewards are calculated in each location of the figure.

react to products and how they are trained to make better decisions. Figure 6.15 shows how various boundaries $\mathbf{b}_{f,l}$, influence consumer decisions while training. The Y axis represents three products labelled with 1, 2 and 3. These products contain different levels of sugar set at 0.5, 0.3 and 0.1. The x-axis represents a consumer's boundaries where sugar values are set to high, medium and low. All consumers' weights are set to 1. Following Equations 6.33, 6.34 and 6.35, the boundaries of consumers are derived. The reward of an action can be determined by Equation 6.38. From the rewards $r_{i,l}$ in Figure 6.15, consumer agents with different boundaries receive different rewards. For example, a consumer agent with a low sugar boundary on the third column receives a positive value of reward only if he or she selects the third product. Any other decisions will turn into punishment, which is a negative value of reward. Another example from the product perspective is product2 in the second row. Product2 provides a positive reward to the agent with a high sugar boundary, zero reward to the agent with a medium sugar boundary and a negative reward to those with a low sugar boundary. As a result, for an agent with a high sugar boundary, it is beneficial to choose product2. Therefore, during training the agent model receives positive rewards when making this decision. On the other hand, for an agent with a low boundary, it is bad for them to choose product2. Therefore during training, the agent model receives training with punishment when making this type of decision.

Next, I experiment with weight $\mathbf{W}_f$ while fixing the boundary values at 0.5, as shown in Figure 6.16. In this experiment, the x-axis records consumer agents with different weights W_f. The first consumer agent's sugar weight in the second column is set to 1.5, while all the other weights are set to 1. All of the second consumer agent's weights are set to 1. The third consumer agent's taste weight is set to 1.5 and his or her weights are set to 1. The products in the y-axis contain different features $P_{f,i}$. The first product in the second row contains sugar at a value of 0.7, taste at a value of 0.3 and other features set at 0.5. The second product has all feature values set at 0.5. The third product contains a sugar value set at 0.3, taste at 0.7 and all other features at 0.5. The key comparison for this experiment is to observe agents 1 and 3 being trained when choosing product1 and product3. Agent 1 receives a positive reward when choosing product1 and a negative rewards when choosing product3. Agent 3 has an inverse reward structure when choosing between products 1 and 3. The different reward structure is due to the application of different weights. Agent 1 tends to emphasise sugar. Therefore during the trade off between the triple constraints shown in Figure 6.7, sugar as a health feature is 1.5 times more effective than the taste feature. Following Equations 6.37 and 6.38, this results in a negative reward when selecting product1. For agent 3, the situation is the opposite due to the fact that he or she emphasises taste 1.5 times more than other features. By differentiating the weights, a different optimal personalised decision is created. Through training the agent model with contrasting reward systems, the agents behave differently.

In summary, the combination of boundary $\mathbf{b}_{f,l}$ and weight $\mathbf{W}_f$ influences how consumer agents are trained. The boundary controls the direction of training, while weight controls the emphasis of features. Changes in these parameters facilitate the observation of different desired behavioural outcomes.

Once consumer agents are trained and ready to react to products, they are applied to train manufacturer agents. In order to make my designed simulation realistic, I replaced the randomly generated training product features set with that generated by manufacturer agents. I took the same framework from Figure 6.9 and designed the shopping environment in detail as shown in Figure 6.17.

The products' features are first initialised using Equation 6.13 and expanded depending on general dietary or personalised dietary parameters following Equation 6.16 or 6.17. The product features are then sent to consumer agents so that they can make decisions as to which products to purchase.

The consumer decisions are summarised into a sales record following Equation 6.49, where c is the consumer label in consumer list CF and i is the identifier of a product label in product list PF, as explained before.

Features in z-score x=1 / Product features \ Benefit boundary of Consumer	W_{sugar}=1.5 W_{other}=1 $b_{all.medium}$ =0.5	W_{all}=1 $b_{all.medium}$ =0.5	W_{taste}=1.5 W_{other} =1 $b_{all.medium}$ =0.5
Product1: $p_{sugar,1}$ = 0.7 $p_{taste,1}$ = 0.3 $p_{other\ features,1}$ = 0.5	$r_{1,medium}$ = -0.1	$r_{1,medium}$ = 0	$r_{1,medium}$= 0.1
Product2: $p_{sugar,2}$ = 0.5 $p_{taste,2}$ = 0.5 $p_{other\ features,2}$ = 0.5	$r_{2,medium}$ = 0	$r_{2,medium}$ = 0	$r_{2,medium}$=0
Product3: $p_{sugar,3}$ = 0.3 $p_{taste,3}$ = 0.7 $p_{other\ features,3}$ = 0.5	$r_{3,medium}$ = 0.1	$r_{3,medium}$ = 0	$r_{3,medium}$= -0.1

Fig. 6.16 The second consumer experiment with different nutritional boundaries setup. The vertical axis presents various products setup and the horizontal axis presents different benefit boundaries of consumers. The rewards are calculated in each location of the figure.

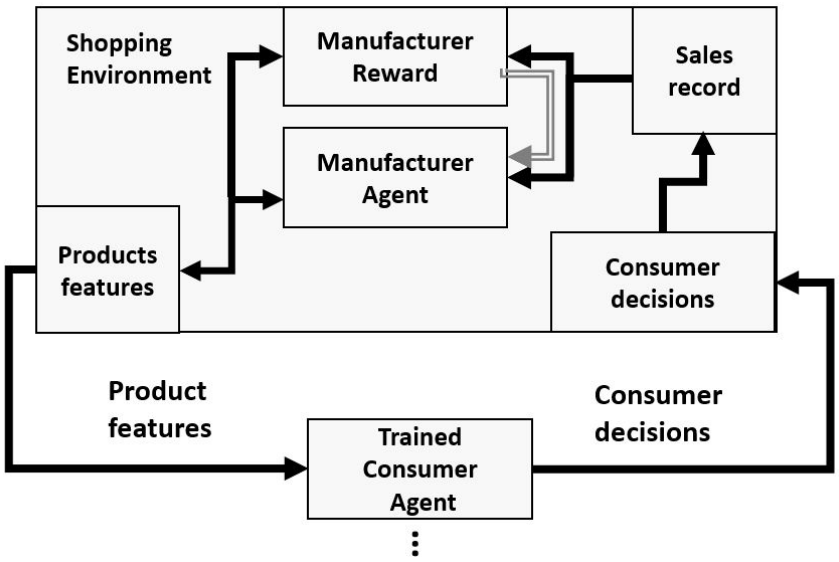


Fig. 6.17 The manufacturer behaviour simulation agent model is trained by consumer agents. The state of a manufacturer is defined by the sales record derived from consumer decisions and the action is equivalent to modifying a product feature. The reward is calculated from sales and profit improvements.

$$p_{sales,i} = \sum_{c=1}^{C} a_{i,c} \tag{6.49}$$

The manufacturers are also exposed to Cost and Profit features. The Cost is used to describe the expense of producing the product, denoted as $p_{cost,i}$. The Profit is expressed as $p_{profit,i}$ which is derived in Equation 6.50.

$$p_{profit,i} = (p_{price,i} - p_{cost,i}) * p_{sales,i} \tag{6.50}$$

Finally the manufacturers are set to observe the business related features and relative product features, see Equation 6.51. The business related features are further expressed as profit, price, cost and sales, as in Equation 6.52 while the relative product features are further expressed as taste and health as shown in Equation 6.53.

$$\mathbf{o}_{manufacturer,i} = [p_{businessfeature,i}, p_{productfeature,i}] \tag{6.51}$$

$$p_{businessfeature,i} = [p_{profit,i}, p_{price,i}, p_{cost,i}, p_{sales,i}] \tag{6.52}$$

$$p_{productfeature,i} = [Mean(\mathbf{q}_{taste}) - p_{taste,i}, Mean(\mathbf{q}_{health}) - p_{health,i}] \tag{6.53}$$

The manufacturer agents have the same structure as the consumer agent in Figure 6.12. The input of manufacturer agents is connected to an observation vector $\mathbf{o}_{manufacturer,i}$, see equation 6.54.

$$\mathbf{Input}_{manufacturer,i} = \mathbf{o}_{manufacturer,i} \tag{6.54}$$

Given that the DNN structure of the manufacturer is the same as that of the consumers', Equation 6.44 is also responsible for producing an output action of manufacturer $\mathbf{a}_m$. These actions $\mathbf{a}_m$ include increasing or decreasing the following features: price, health and cost of the products, as shown in Equation 6.55. For each time step, only one action element can be one while the rest of action elements are zero.

$$\mathbf{a}_m = [a_{price}^{increase}, a_{health}^{increase}, a_{cost}^{increase}, a_{price}^{decrease}, a_{health}^{decrease}, a_{cost}^{decrease}, a_{nothing}] \tag{6.55}$$

With Equation 6.13, the product feature is modified to include the results after the manufacturers' actions, see Equation 6.56. Where MU is the modification unit that influences the number of modifications taken by manufacturers at each time step.

$$p_f^{new} = p_f + MU * a_f^{increase} - MU * a_f^{decrease} \tag{6.56}$$

Once the manufacturers have modified their products, consumer agents deal with the new modified product feature set $\mathbf{P}_i^{new}$. They then make new decisions resulting in new sales and profits, following Equation 6.49 and 6.50. The new sales and profits are defined as p_{sales}^{new} and p_{profit}^{new} and the original sales and profits are defined as p_{sales}^{old} and p_{profit}^{old}.

Finally, the rewards to the manufacturer agents are derived by the sum of profit improvement (PI) and sales improvement (SI), see Equation 6.57. PI is the difference between the new and old profits multiplied by the old sales, while SI is the difference between the new and old sales multiplied by the old profits. These are defined in Equations 6.58 and 6.59.

$$r_{manufacturer} = PI + SI \tag{6.57}$$

$$PI = (p_{profit}^{new} - p_{profit}^{old}) * p_{sales}^{old} \tag{6.58}$$

$$SI = (p_{sales}^{new} - p_{sales}^{old}) * p_{profit}^{old} \tag{6.59}$$

Following the same concept as Equation 6.45, the expected action of manufacturer $a_m^{expected}$ is defined as y_m as shown in Equation 6.60.

$$\mathbf{y}_m = \mathbf{a}_m^{expected} \tag{6.60}$$

The expected action of the manufacturer is derived from manufacturers' previous actions $\mathbf{a}_m$ and their rewards $r_{manufacturer}$ as in Equations 6.61 and 6.62.

$$\mathbf{a}_m^{expected} = \mathbf{a}_m + \mathbf{r}_m^v = [a_1, a_2, ..., a_i, ..., a_I]_m + [0, 0, r_{gm}, ..., 0]_m \tag{6.61}$$

$$\mathbf{r}_m^v = [0, 0, r_{gm}, ..., 0]_m = r_m \mathbf{u}_{gm} \tag{6.62}$$

In Equation 6.62, $\mathbf{u}_{gm}$ is a directional unit vector that points in the gm direction, where $gm = \mathbf{argmax}(\mathbf{a}_m)$ is the location of the action that the agent decides to perform.

Following the same concept of Equation 6.48, while considering future states, the expected action of manufacturers can be expanded as in Equation 6.63. Where the confidence factor of a manufacturer is expressed as $\mathbf{f}_m = max(\mathbf{a'}_m)\mathbf{u}_{gm}$.

$$\mathbf{a}_m^{expected} = \mathbf{a}_m + \mathbf{r}_m^v + \gamma \mathbf{f}_m \tag{6.63}$$

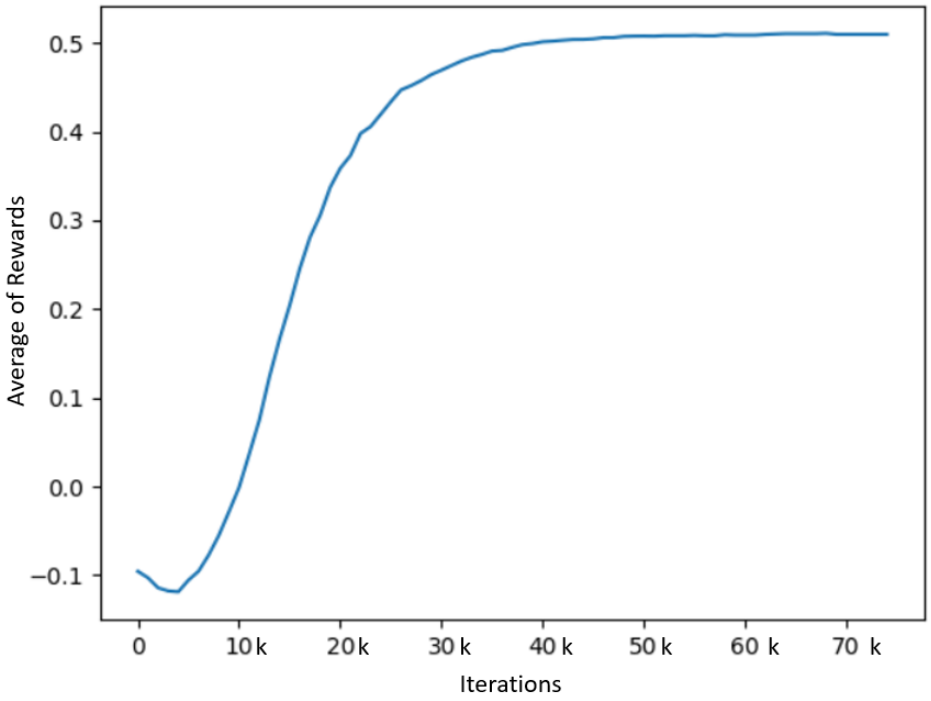


Fig. 6.18 Average reward of manufacturer agents training. The horizontal axis represents the training step iterations.

With the same procedure as the consumer agents, the manufacturer agents memorise multiple sets of $[\mathbf{s}_m, \mathbf{a}_m, \mathbf{r}_m, \mathbf{s'}_m]$, following Equation 6.64. The manufacturer agents are trained by looping through the history records. The Stochastic Gradient Descent (SGD) method is applied to optimise the model updates.

$$[\mathbf{s}_m, \mathbf{a}_m, \mathbf{r}_m, \mathbf{s'}_m] = [\mathbf{o}_{m,i}, \mathbf{a}_m, \mathbf{r}_m, \mathbf{o'}_{m,i}] \tag{6.64}$$

The manufacturer agents model is trained through feeding randomised observation inputs. They receive rewards when a good action is chosen. The randomised observations vary from 50% to 150% of the average value of existing products. The unit of modification MU is set to 0.1. The test data is generated before training, containing 1000 randomly generated products.

In each iteration, the model is updated based on the rewards due to the interaction between the manufacturer agents and the consumer agents. Once the model has been trained, it is used to perform actions on the testing sets. The rewards of actions from the testing sets are collected and observed as shown in Figure 6.18. The maximum average reward is 0.5, which is achieved after 57100 iterations.

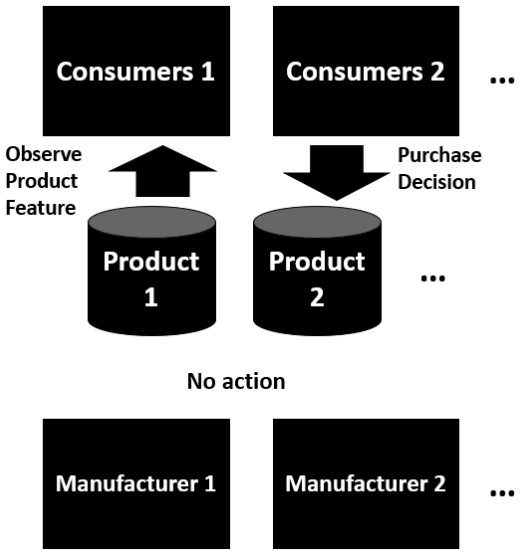


Fig. 6.19 A simulation experiment that focuses on consumer and product interactions without considering manufacturers. In this experiment, consumers are set to make decisions on products they prefer to purchase while manufacturers are not allowed to modify products' features.

6.3 Simulation Outcome and Discussion

In this section, the consumer agent model and the manufacturer agent model are now ready to construct the G-BUSiNESS simulation on existing datasets, such as supermarket websites. I draw conclusions about the benefits of these simulations using different assumptions and setups.

I begin by extracting 40 products from the cookie category on the Tesco supermarket website. Information about each product and its price, energy, fat, saturated fat, sugar and salt are recorded. The taste feature of each product is also included following Equation 6.18. The cost of each product is assumed to be half of its price.

After setting up the initial states of products, three simulations with different variations are conducted: variation of consumer models, variation of manufacturer models and variation of the whole environment.

In the first simulation, the product features are fixed and consumers are free to make decisions on purchasing one product at each time step, as shown in Figure 6.19. With this setup, consumer decisions can be observed and summarised. Consumers are partitioned into two main groups: a personalised diet recommendation group and a generalised recommendation group. Within each group, they are further partitioned into smaller groups that target different dietary requirements such as lower sugar intake from the purchased products. This experiment

Table 6.1 The average value of product features that consumers choose to purchase. The vertical axis represents different groups of consumers, and the horizontal axis represents various product features.

	Price	Energy	Fat	Saturated fat	Sugar	Salt
Random	0.84	2069.55	24.71	12.80	32.85	0.59
Personalised (all)	0.51	1925.23	21.61	10.81	21.81	0.40
Personalised (Low Sugar group)	0.43	1892.67	22.77	11.63	**9.37**	0.42
Generalised (all)	**0.38**	2019.75	22.99	11.38	28.00	0.38
Generalised (Low Sugar group)	**0.38**	2008.87	22.84	11.31	26.89	0.39

Table 6.2 The average value of product features in percentage that consumers choose to purchase. The vertical axis represents different groups of consumers while the horizontal axis represents various product features.

	Price	Energy	Fat	Saturated fat	Sugar	Salt
Random	100%	100%	100%	100%	100%	100%
Personalised (all)	61%	93%	87%	84%	66%	68%
Personalised (Low Sugar group)	51%	91%	92%	91%	**29**%	72%
Generalised (all)	**45**%	98%	93%	89%	85%	65%
Generalised (Low Sugar group)	**45**%	98%	93%	89%	85%	65%

aims to study the variations in consumer agents' decisions due to different exposure to product information.

Three experiments are conducted here to observe the impacts on price, energy, fat, saturated fat, sugar and salt, when three kinds of purchase decisions are made by consumers: 1) **Random decisions:** agents choose products randomly, while all products have the same probability to be chosen. This is considered to be a relative baseline. 2) **Personalised agent decisions:** agents make decisions after observing each products' price, taste and health factor. The average number of product features for all groups focusing on different features is considered. Particular focus will be given to decisions made by agents in the lower sugar intake group. 3) **Generalised agent decisions:** agents make decisions after observing the products' price, taste and the average of all health factors. The average number of product features for all the groups focusing on different features is considered. Particular focus will be given to decisions made by agents in the lower sugar intake group. The results are shown in Table 6.1. The values in Table 6.1 are divided by the values of the random decisions row and are displayed as percentages in Table 6.2 and Figure 6.20.

In Figure 6.20, both personalised and generalised agents make better decisions regarding products' features compared to agents making random decisions. Specially, the price drops significantly for both agents. Decisions made by personalised agents result in lower sugar

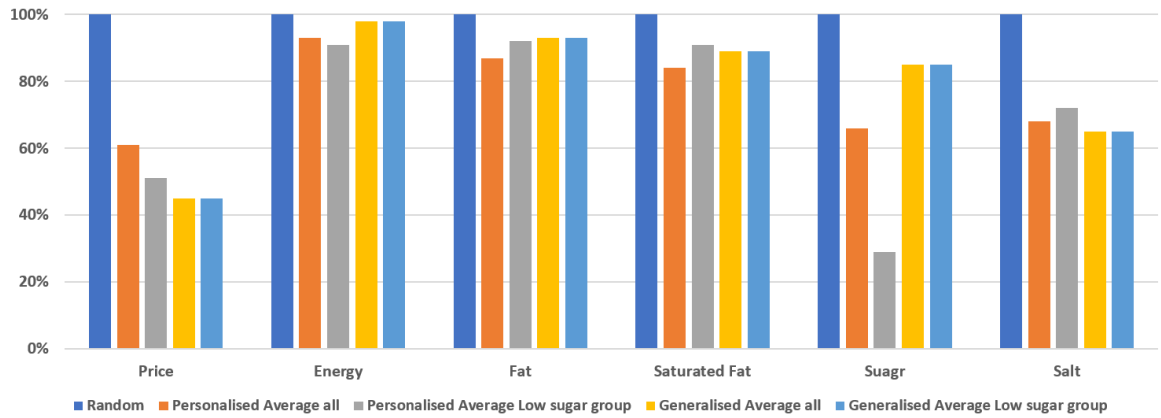


Fig. 6.20 The average value of product features as percentages that consumers choose to purchase considering different grouping criteria. The vertical axis represents the value of a feature selected by a group of consumers divided by the value of a feature of a randomly selected product. The different groups of consumers are presented in different colours while the horizontal axis represents various product features.

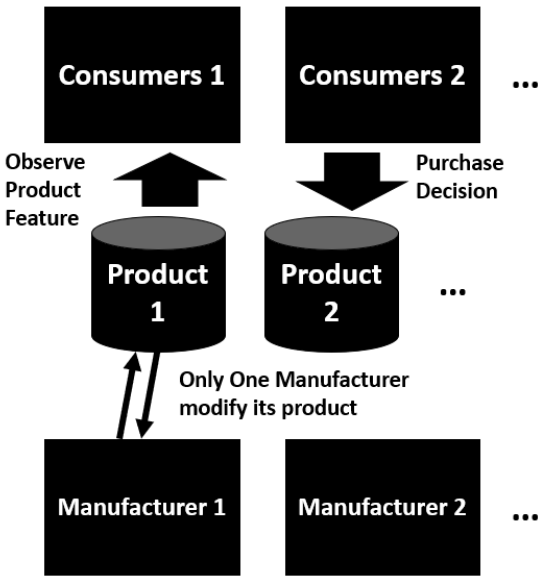


Fig. 6.21 A simulation of consumer decisions with only one manufacturer modifying its product features in each simulation. In this experiment, consumers are set to make decisions on products they prefer to purchase while one of the manufacturers is set to modify the products' features.

intake than decisions made randomly or by generalised agents. In particular, the actions of the smaller personalised agent group, aiming to lower sugar intake results in a much lower sugar intake while trading off increased salt and saturated fat intakes.

These experiments show that decisions made by generalised consumer agents result in lower prices and lower nutrition intakes. Decisions made by personalised consumer agents result not only in lower prices, but also in much lower nutrition intake. In particular, for the smaller groups aiming to reduce a specific feature, consumer agents make decisions that lead to a much lower intake of the required feature without trading off too much of the other features.

In order to investigate the strategies of the trained manufacturer agents in reacting to the simulated market, I modify the features of one product by one agent at each time step and fix the features of others. As shown in Figure 6.21, this variation aims to experiment with the change that occurs when only one product is improved with its features by the manufacturer agent while its competitors do not make any modifications. In other words, I want to simulate a scenario whereby only one manufacturer works with the personalised service, in order to determine how much optimisation can be achieved. I conducted 40 simulations with 40 individual worlds, each based on 40 products.

Figure 6.22 shows the behaviour of a typical personalised manufacturer agent. After the 43rd iteration, the personalised agent starts to adjust the direction of optimisation. The cost feature drops because the decreasing health feature affects the taste feature, which results in a decrease in sales. After several iterations, the agent determines a better solution through trading off sales and costs. After the 59th iteration, the agent ceases the modification process at its optimal point. Achieving this point means that the agent has reached saturation and has become stable. Overall, the most important action that a personalised agent tends to make and optimise is to decrease nutrition features such as salt, sugar and more. The health feature in this figure is the averaged nutrition intake, where less health means less sugar, salt and so forth, which is better for customers. Another important feature is decreasing price, which might attract more customers and result in an increase in sales. The personalised agent also increases the cost feature, which results in an improved quality of product and may further boost sales. The increased sales will eventually result in an overall profit increase.

Figure 6.23 shows the behaviour of a typical generalised manufacturer agent. Due to the fact that the generalised consumer agents observe the general nutrition values, the manufacturer agent encounters difficulty in targeting a certain group of consumers' health specifications. Instead, only trading off price and sales proves simpler for a generalised manufacturer agent. Therefore it only decreases price to increase sales until saturation is reached.

The average values of product features from 40 simulations are shown in Table 6.3, and the corresponding percentages are shown in Table 6.4. The price drops are greater in generalised diets compared to personalised diets, while nutrition (health) drops more in personalised diets. The cost is higher in personalised diet simulations. In summary, both manufacturers provide products with lower prices, while personalised manufacturers provide products that are healthier

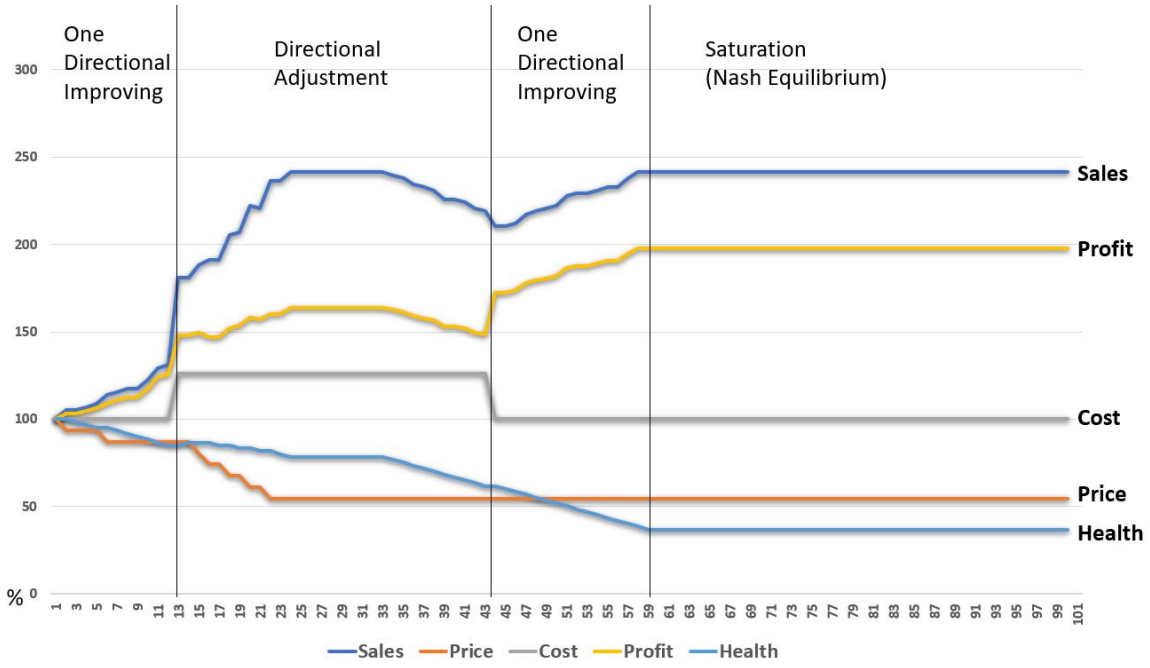


Fig. 6.22 The behaviour of a typical personalised manufacturer agent is shown. The vertical axis represents the variation from the original value in percentage; 100 percent is the original value. The horizontal axis is the time step of the simulation.

Table 6.3 The average value of product features that consumers choose to purchase after manufacturers modify their products. The vertical axis represents different groups of consumers while the horizontal axis represents various product features.

	Price	Cost	Energy	Fat	Saturated fat	Sugar	Salt
Original Features	0.84	0.21	2069.55	24.71	12.80	32.85	0.59
Personalised Diets	0.73	**0.26**	**1977.25**	**23.18**	**11.77**	**31.31**	**0.54**
Generalised Diets	**0.47**	0.21	2050.92	24.71	12.73	32.64	0.58

(lower nutrition intake) and have better quality (higher cost). Specifically, Table 6.5 shows the average profit gain of 40 simulations of different manufacturer agents. Both personalised and generalised diets achieve a higher gain even though they provide products that have higher costs or lower prices.

The third experiment aims to simulate the behaviour of the whole market as a business union consisting of 40 products and 40 manufacturer agents. The top level process is shown in Figure 6.24. The manufacturer agents first observe their corresponding product features, business related factors (such as sales, costs etc.). They modify these features and factors simultaneously to compete with each other. Figures 6.25 and 6.26 show the behaviours of the whole market with personalised manufacturer agents and generalised manufacturer agents.

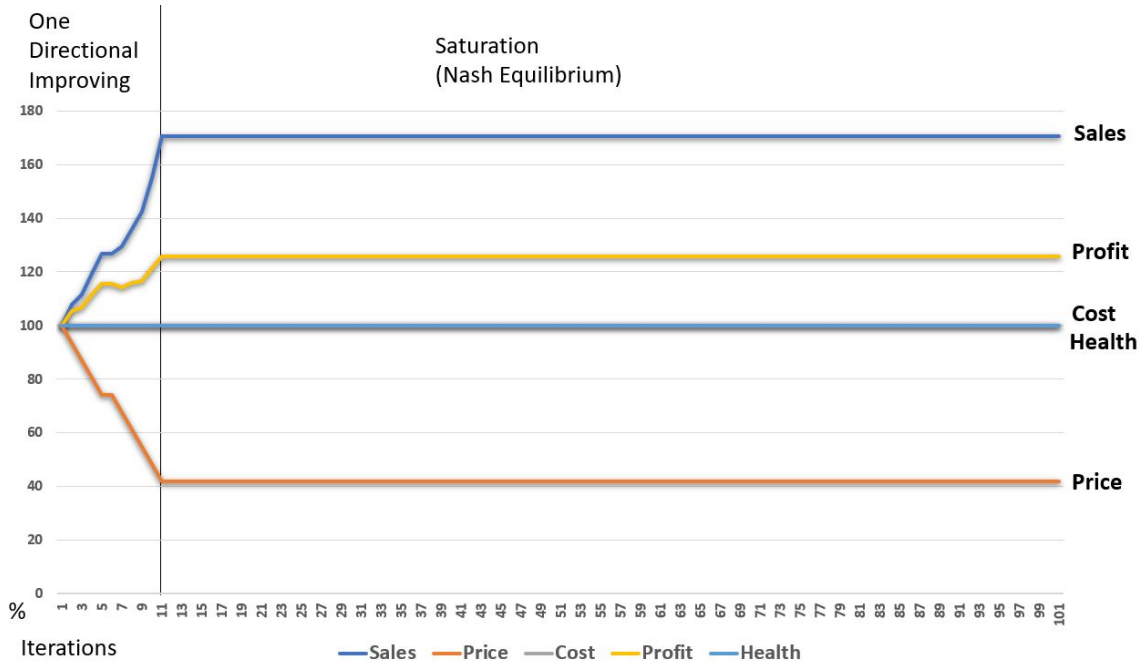


Fig. 6.23 The behaviour of a typical generalised manufacturer agent is shown. The vertical axis represents the variation from the original value in percentage; 100 percent is the original value. The horizontal axis is the time step of the simulation.

Table 6.4 The average value of product features in percentages that consumers choose to purchase after manufacturers modify their products. The vertical axis represents different groups of consumers while the horizontal axis represents various product features.

	Price	Cost	Energy	Fat	Saturated fat	Sugar	Salt
Original Features	100%	100%	100%	100%	100%	100%	100%
Personalised Diets	86.86%	**121.74%**	**95.54%**	**93.79%**	**91.97%**	**95.31%**	**92.39%**
Generalised Diets	**55.6%**	100%	99.1%	100%	99.46%	99.37%	99.21%

Table 6.5 The average profit gain of the business for manufacturers facing different groups of consumers in the simulation. The vertical axis represents different groups of consumers.

	Profit Gain
Original Features	100%
Personalised Diets	**126.89%**
Generalised Diets	123.67%

The behaviours of the whole market with personalised manufacturer agents show similar trends to the previous simulations where there is only a single agent in the market and no competition occurs. The cost continuously increases for all 40 products, while price and health (nutrition such as sugar) decrease (as before, lower health means lower nutrition intake, which

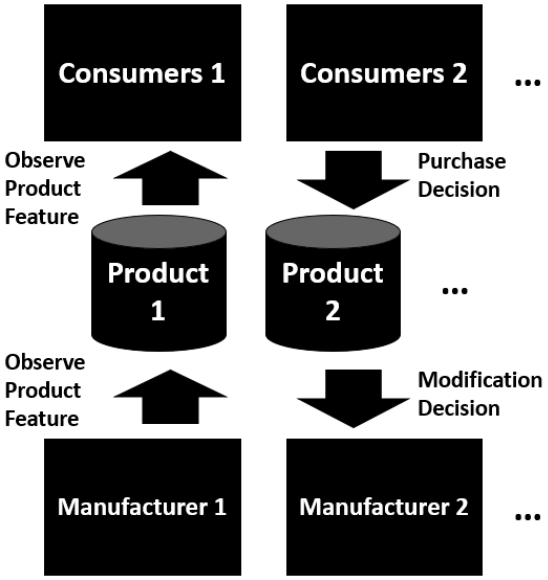


Fig. 6.24 A top-level GBUSiNESS simulation process. In this experiment, consumers are set to make decisions on the product they prefer to purchase while the manufacturers are set to modify products' features in each iteration of the simulation.

means the products are better for consumers' health). However, the profits all decrease due to competition.

The behaviours of the whole market with generalised manufacturer agents show a situation of price wars, where all manufacturers seek to decrease their prices to obtain more sales. The other features are less important. Given that the competition is mainly based on prices, the average profit of the products decreases.

The total number of sales remains constant during the simulation. Therefore, the sales features in Figures 6.25 and 6.26 do not change. However, the distribution and standard deviation change over iterations, as shown in Figure 6.27. The two left figures are the sales distributions before the simulation, where all products have their original features. The two right figures are the sales distributions after the simulation, where all product features are modified. The standard deviations quantify the amount of variation of the sales of all products. Before the simulation, the market with personalised manufacturer agents has a higher standard deviation, which means that the variation of sales over products varies a lot: some products dominate the market share, while others have very little market share. The market with generalised manufacturer agents, it has a smaller standard deviation, which results in a more evenly distributed market share. After the simulation, the market with the personalised manufacturer agents results in a decreased standard deviation, hence the distribution of product sales becomes more even. The negative standard deviation variation also indicates the same distribution. As

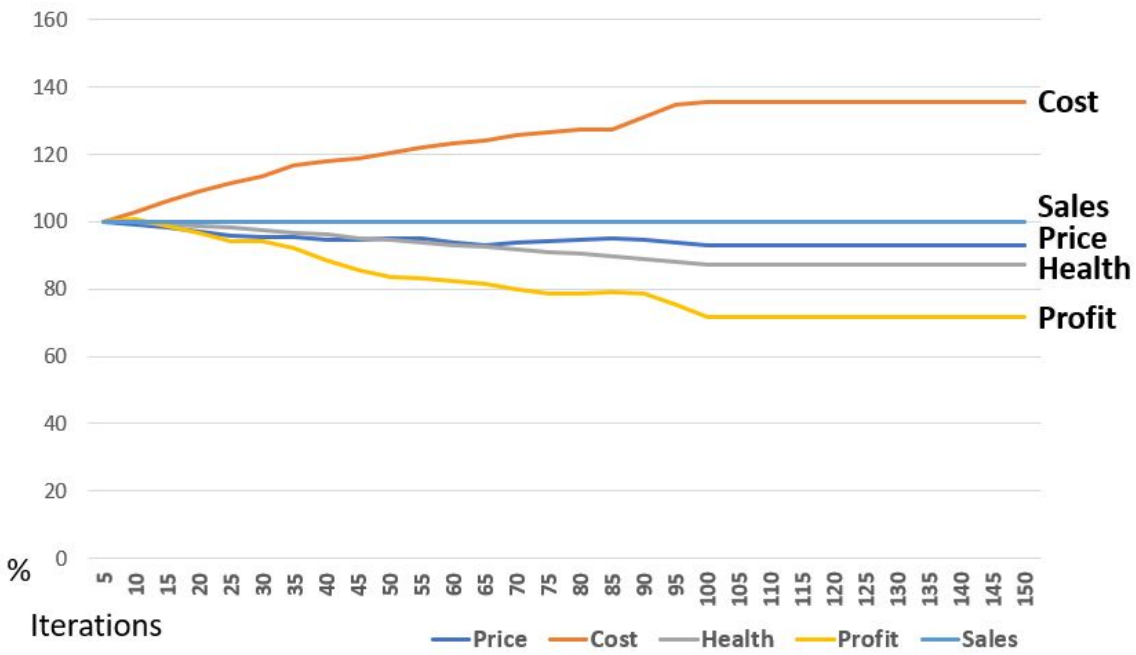


Fig. 6.25 The behaviour of the supermarket with personalised manufacturer agents is shown. The vertical axis represents the variation from the original value in the percentage; 100 percent is the original value. The horizontal axis is the time step of the simulation.

Table 6.6 The average value of product features that consumers choose to purchase after manufacturers modify their products. The vertical axis represents different groups of consumers while the horizontal axis represents various product features.

	Price	Cost	Energy	Fat	Saturated fat	Sugar	Salt
Original Features	0.84	0.21	2069.55	24.71	12.80	32.85	0.59
Personalised Diets	0.78	**0.28**	**1822.67**	**21.59**	**11.25**	**28.83**	**0.50**
Generalised Diets	**0.42**	0.21	1857.66	23.30	11.97	30.62	0.54

for the market with generalised manufacturer agents, it has an increased standard deviation, which results in a less evenly distributed market share. As the simulation with generalised manufacturer agents represents a price war, the increased standard deviation means that some manufacturers win. In this figure, *"SD diff"* represents the difference between standard deviations. The decrease of *"SD diff"* indicates that the market with personalised manufacturer agents achieves their targeted market only by optimising their product features specifically for a certain group of consumers.

Table 6.6 and 6.7 show the average values and percentages of product features from a single simulation with 40 active manufacturer agents. The price decreases significantly in the simulation with generalised diet, while the cost increases significantly in the simulation with

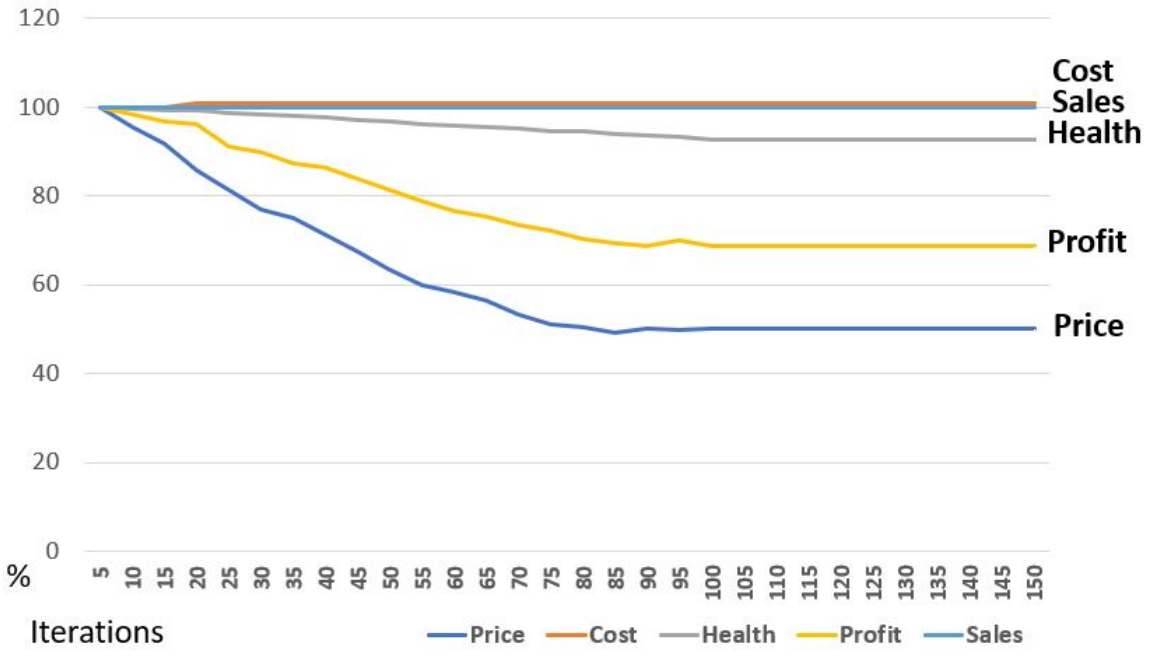


Fig. 6.26 The behaviour of the supermarket with generalised manufacturer agents is shown. The vertical axis represents the variation from the original value in percentage, which 100 percent is the original value. The horizontal axis is the time step of the simulation.

Table 6.7 The average value of product features in percentage that consumers choose to purchase after manufacturers modify their products. The vertical axis represents different groups of consumers while the horizontal axis represents various product features.

	Price	Cost	Energy	Fat	Saturated fat	Sugar	Salt
Original Features	100%	100%	100%	100%	100%	100%	100%
Personalised Diets	93.14%	**135.60**%	**88.07**%	**87.39**%	**87.95**%	**87.75**%	**85.56**%
Generalised Diets	**50.30**%	100.91%	89.76%	94.30%	93.57%	93.22%	92.37%

personalised diet. The health features (including energy, fat, saturated fat, sugar and salt) in the personalised diet simulations decrease more than in the generalised diet simulations. This is a similar outcome to the single agent simulation discussed in the previous simulation, which is an environment without competition. However, the profit gains are different in this simulation: dropping in both the personalised diet and the generalised diet simulations. These results are in fact to be expected due to the competition in this simulation. In particular, the profit gain of the personalised diet drops slightly more than in the generalised diet due to the higher cost in the former simulation.

These three simulations confirm the advantages of personalised diets for consumers in terms of purchasing better quality (higher cost) and healthier (lower nutrition, such as sugar, salt and

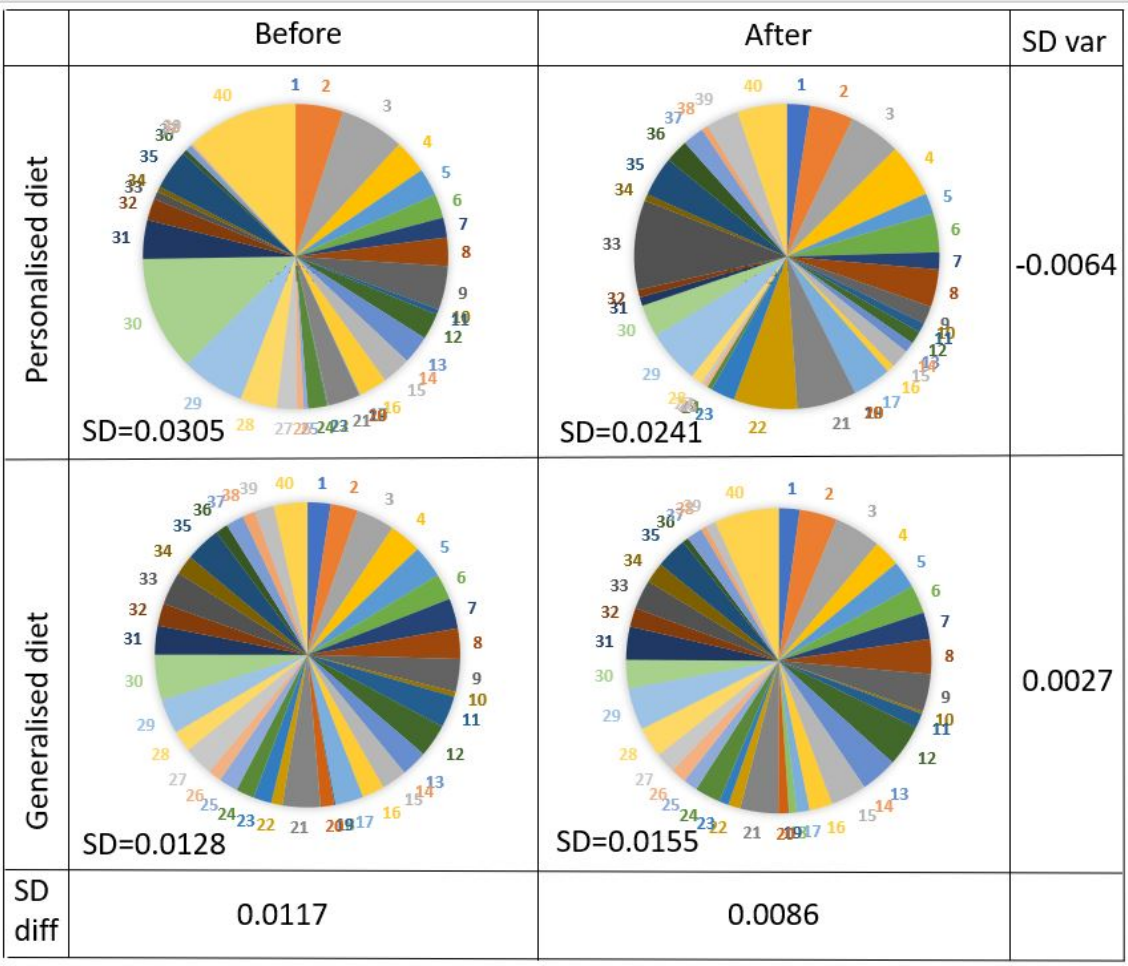


Fig. 6.27 Sales distribution and standard deviation of the market with personalised and generalised manufacturer agents.

Table 6.8 The average profit gain of the business for manufacturers facing different groups of consumers in the GBUSiNESS experiment. The vertical axis represents different groups of consumers.

	Profit Gain
Original Features	100%
Personalised Diets	**68.88**%
Generalised Diets	71.69%

etc) products. In lower or non-competitive environments, manufacturers with recommendations from our service receive higher profits. In the third simulation, due to competition, supermarkets with PERSON have better quality products while manufacturers compromise the profits slightly. The simulation models can potentially be applied to marketing strategic planning and prediction for retailers or supermarkets, manufacturers and even government. The first simulation with

consumer agents suggests personalised recommendations based on genes can help consumers to choose products that are better and healthier for them. For example, consumers with low sugar tolerance genes will choose cookies with low sugar, as shown in Table 6.1. The second simulation with a single manufacturer agent demonstrates the benefit when a manufacturer modifies its product and obtains more profit. Specifically, when it modifies its product features with personalised diets, it can gain more profit while producing products that are healthier (lower energy, fat, sugar and salt), and of better quality (higher cost). However, the second simulation only has one manufacturer working inside my designed system and changing its product features (without competition). In the third simulation, I consider the whole market as a business union by including competitions between 40 products and 40 manufacturer agents. Each manufacturer makes less profit due to the competition. However, consumers receive better quality and healthier products when they have access to personalised diet recommendations based on their genes. The consumer agent model and the manufacturer agent model described here can be used in real-world applications with real-world data.

Chapter 7

Conclusion

In this chapter, I first summarise of each chapter and the three original contributions in Section 7.1. The factors that can potentially be explored further in the future are described in Section 7.2. Finally, I present my final thoughts of this research in Section 7.3.

7.1 Original Contributions

In this PhD research, I aim to minimise the gap between DNA information and daily consumer usage through an expert system that implements an application of Nudge theory. I have published a description of my proposed expert system for DNAnudge in a journal paper entitled "PERSON-Personalised Expert Recommendation System for Optimised Nutrition" in *IEEE Transactions on Biomedical Circuits and Systems* [22]. A report on the same system has also been published as a chapter in the book *Trends in Personalized Nutrition 1st Edition*, published by Elsevier on 24 May 2019 [44].

In Chapter 3, I presented a review of existing research on automatic data collection, and described how the data for my work were collected. I also described the method by which the data was stored, processed and used by the system PERSON. In addition, in order to implement PERSON in real-world systems, I discussed its scalability. In Chapter 4, the product names were converted from human language to machine representation, i.e. from English text to numerical vectors. To this end, I applied a generalised multi-million-word dictionary model called Skip-Gram Bag-of-Words, trained using an unsupervised approach on Google News datasets. With this model, my designed system was able to scale up to cover not only manually created training datasets (supermarket products datasets) but also several millions other words. To optimise the accuracy of the categorisation, I experimented with and compared several different machine learning models. Based on their accuracy, I selected a type of Recurrent Neural Network called LSTM for implementation in my designed system. In Section 4.4, I described a novel

recommendation model that compares the nutritional values of products grouped into categories by my chosen machine learning model. This recommendation was optimised using a genetic algorithm. At the end of the chapter, the potential reductions in unwanted nutritional intake were presented for various food categories, based on our collected data and the application model. In Section 4.5 I combined the different components to explain the operation of my designed expert system and its outcomes. This system links DNA information with food product decisions by considering the optimisation of nutrition, and demonstrates the power of nudging a consumer from one product towards a healthier one in the same category. For example, from buying chocolate containing more sugar to a healthier chocolate with less sugar. These recommendations are personalised based on each person's genetic information through identifying a trade-off in terms of the priorities of different nutrients. Using a simple behaviour simulation that considers the probability that a consumer will follow recommendations, I highlighted how much unwanted nutritional intake can be reduced. In Chapter 5, The proposed PERSON system was applied in real-world markets. In this trial, the influences of Self-nudge and DNAnudge were discussed. Based on the real-world data collected from the trial and application usage records, a mathematical model was constructed to represent the immediate and memory effects of DNAnudge.

Finally, due to the fact that this new system is still in its early stages and has not yet been used in real-world applications, the available data is not sufficient to measure the influence of my proposed system on the market. However, I explored this potential influence in Chapter 6. Using a realistic simulation grounded in a behavioural game theory environment, I studied consumer, retailer and manufacturer interactions and measured the influence of the system. In this simulation, PERSON was used to generate personalised recommendations to customer agents. I also compared it with generalised recommendations and no recommendations, resulting in a better business strategy for customers, manufacturers and retailers. Three key inventions and contributions have been made here, which can be summarised as follows:

1. The application consists of a categorisation model and a recommendation model: In Chapter 4, I described my invented framework, PERSON, which performs personalised grocery product recommendations based upon customers' genes.

2. Trial and influence demonstration: I demonstrate the influences of Self-nudge and DNAnudge in Section 5 through conducting a trial in a supermarket. The collected data was explained through a mathematical model.

3. Simulation model: I created a simulation framework of the business union to simulate the behaviours of consumers and manufacturers following recommendations provided by

PERSON. This framework brings PERSON into the business environment by providing decision assistance for product manufacturers, as outlined in Section 6.

7.2 Future Works

In this PhD thesis, I have demonstrated that healthier food can be chosen by nudge using personalised product selections. In the PERSON framework introduced in Chapter 4, I applied five nutrition-genes mapping. The work with the team at DNAnudge has been focused on increasing the number of nutrition-genes mapping with consideration of life style factors. This will help modify recommendations to give a more in-depth personalisation and boost the accuracy of recommendations to meet consumers' needs. From a trial conducted at a supermarket and retail chain involving the PERSON framework introduced in Chapter 5, I concluded that the more frequent the users of the DNAnudge app, the longer their memories of suitable products to be chosen. This trial can be extended by exploring the reasoning behind behavioural changes. One approach can be to review of consumers' shopping history before swapping a decision from buying a not recommended product to a recommended product. Such reasoning can potentially be valuable for marketing strategy design. The PERSON servicing framework has now been launched into the real-world environment. Nonetheless, the completed works can only cover business-to-consumer (B2C) servicing. B2C servicing is usually less profitable because it has to trade off prices against the number of users. In other words, if the service becomes more expensive, fewer consumers will purchase and use it. It is our goal at DNAnudge to scale up the number of users and to create intelligence on top of users' data to consult other business organisations in order to achieve better health, sales, and production considering personalisation. My research shows that generating recommendations that can influence retailers and manufacturers to achieve better production of food through simulation can bring positive impacts. Research into finding a valuable business-to-business (B2B) servicing model remains ongoing at DNAnudge. The simulation introduced in Chapter 6, GBUSiNESS is one example of a B2B consultation service that aims to find an optimisation strategy for modifying ingredients and pricing. This framework is built on the basis of assumptions and can later be modified to adapt to real-world data. In my future work, I aim to replace each assumption made in GBUSiNESS with individual models that mimic each behaviour created on the basis of the user data collected. The power of DNAnudge can be extended through collecting real-world behavioural data of consumers from the PERSON service. Bringing the simulation system closer to reality is likely to have a greater impact on the food industry.

7.3 Final Thoughts

I believe that DNAnudge is an effective solution for incorporating DNA information into everyday decision-making processes. To the best of my knowledge, it represents one of the best commercial applications for demonstrating the benefit of Nudge Theory based on genetics. It is expected that this research of DNAnudge on grocery food will benefit not only consumers but also retailers and manufacturers. Through future research on the extension of the PERSON service, it will be more finely tuned and constantly updated to help consumers find better food products, while the GBUSiNESS framework digests real-world data and predicts consumer behaviours, which can help retailers and manufacturers with business strategy simulations. Aside from my future work described in the previous section, it is possible to reuse parts or the whole of my invented framework structure for recommending products beyond food, such as cosmetics, supplements, sports equipment, fragrances, and more. The goal of my research is to nudge the food industry towards personalised, healthier decisions. However, the goal of DNAnudge is ultimately to improve the majority of our everyday decisions through nudge by considering the source code of humans, DNA. I am honoured to be the first person to invent an application to demonstrate DNAnudge.

References

[1] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.

[2] Agarwal, N. and Liu, H. (2008). Blogosphere: research issues, tools, and applications. *ACM Sigkdd Explorations Newsletter*, 10(1):18–31.

[3] Agarwal, N. and Zeephongsekul, P. (2011). Psychological pricing in mergers & acquisitions using game theory. *School of Mathematics and Geospatial Sciences, RMIT University, Melbourne*.

[4] Aizerman, M., Braverman, E. M., and Rozonoer, L. (1964). Theoretical foundations of potential function method in pattern recognition. *Automation and Remote Control*, 25(6):917–936.

[5] Amari, S.-I. (1990). Mathematical foundations of neurocomputing. *Proceedings of the IEEE*, 78(9):1443–1463.

[6] Angers, B., Castonguay, E., and Massicotte, R. (2010). Environmentally induced phenotypes and dna methylation: how to deal with unpredictable conditions until the next generation and after. *Molecular Ecology*, 19(7):1283–1295.

[7] Anselma, L., Mazzei, A., and De Michieli, F. (2017). An artificial intelligence framework for compensating transgressions and its application to diet management. *Journal of Biomedical Informatics*, 68:58–70.

[8] Bellman, R. (1957). A markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684.

[9] Bengio, Y. et al. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

[10] Bewersdorff, J. and Kramer, D. (2005). *Luck, logic, and white lies: the mathematics of games*. AK Peters Wellesley, Mass.

[11] Bilbao, J. M. (2012). *Cooperative games on combinatorial structures*, volume 26. Springer Science & Business Media.

[12] Bloss, C. S., Schork, N. J., and Topol, E. J. (2011a). Effect of direct-to-consumer genomewide profiling to assess disease risk. *N Engl J Med*, 2011(364):524–534.

[13] Bloss, C. S., Schork, N. J., and Topol, E. J. (2011b). Effect of direct-to-consumer genomewide profiling to assess disease risk. *New England Journal of Medicine*, 364(6):524–534.

[14] Bondi, A. B. (2000). Characteristics of scalability and their impact on performance. In *Proceedings of the 2nd international workshop on Software and performance*, pages 195–203. ACM.

[15] Brandenburger, A. (2007). Cooperative game theory: Characteristic functions, allocations, marginal contribution. *Stern School of Business. New York University*, 1:1–6.

[16] Cahill, L. and El-Sohemy, A. (2011). 4.57-nutrigenomics: A possible road to personalized nutrition. *Elsevier BV*.

[17] Campbell-Arvai, V., Arvai, J., and Kalof, L. (2014). Motivating sustainable food choices: The role of nudges, value orientation, and information provision. *Environment and Behavior*, 46(4):453–475.

[18] Cao, L. (2010). In-depth behavior understanding and use: the behavior informatics approach. *Information Sciences*, 180(17):3067–3085.

[19] Cariaso, M. and Lennon, G. (2011). Snpedia: a wiki supporting personal genome annotation, interpretation and analysis. *Nucleic acids research*, 40(D1):D1308–D1312.

[20] Caulfield, T., Ries, N., Ray, P., Shuman, C., and Wilson, B. (2010). Direct-to-consumer genetic testing: good, bad or benign? *Clinical genetics*, 77(2):101–105.

[21] Chang, C.-H. and Lui, S.-C. (2001). Iepad: information extraction based on pattern discovery. In *Proceedings of the 10th international conference on World Wide Web*, pages 681–688. ACM.

[22] Chen, C.-H., Karvela, M., Sohbati, M., Shinawatra, T., and Toumazou, C. (2017). Person—personalized expert recommendation system for optimized nutrition. *IEEE transactions on biomedical circuits and systems*, 12(1):151–160.

[23] Cherkas, L. F., Harris, J. M., Levinson, E., Spector, T. D., and Prainsack, B. (2010). A survey of uk public interest in internet-based personal genome testing. *PloS one*, 5(10):e13473.

[24] Cheung, T. T., Kroese, F. M., Fennis, B. M., and De Ridder, D. T. (2017). The hunger games: Using hunger to promote healthy choices in self-control conflicts. *Appetite*, 116:401–409.

[25] Chevalier-Roignant, B. and Trigeorgis, L. (2011). *Competitive strategy: Options and games*. MIT Press.

[26] Christen, M. (1998). *Information acquisition by firms: the role of specialization, motivation and ability*. INSEAD.

[27] Chu, Y.-C., Hsu, C.-C., Lee, C.-J., and Tsai, Y.-T. (2015). Automatic data extraction of websites using data path matching and alignment. In *Digital Information Processing and Communications (ICDIPC), 2015 Fifth International Conference on*, pages 60–64. IEEE.

[28] Churchill, F. B. (1974). William johannsen and the genotype concept. *Journal of the History of Biology*, 7(1):5–30.

[29] Clemons, E. K. (2008). How information changes consumer behavior and how consumer behavior determines corporate strategy. *Journal of management information systems*, 25(2):13–40.

[30] Cohen, H. and Lefebvre, C. (2005). *Handbook of categorization in cognitive science*. Elsevier.

[31] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

[32] Cooke, R. L. (2011). *The history of mathematics: A brief course*. John Wiley & Sons.

[33] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

[34] Daniel, H. and Klein, U. (2013). Nutrigenetik: Genetische varianz und effekte der ernährung. In *Biofunktionalität der Lebensmittelinhaltsstoffe*, pages 7–16. Springer.

[35] database, U. (2017). Us department of agriculture, agricultural research service, nutrient data laboratory. usda branded food products database. *USDA database*.

[36] Dogan, I. and Güner, A. R. (2015). A reinforcement learning approach to competitive ordering and pricing problem. *Expert Systems*, 32(1):39–48.

[37] Durlauf, S. N., Blume, L., et al. (2008). *The new Palgrave dictionary of economics*, volume 6. Palgrave Macmillan Basingstoke.

[38] Ebert, P. and Freibichler, W. (2017). Nudge management: applying behavioural science to increase knowledge worker productivity. *Journal of Organization Design*, 6(1):4.

[39] El-Rewini, H. and Abd-El-Barr, M. (2005). *Advanced computer architecture and parallel processing*, volume 42. John Wiley & Sons.

[40] Eng, C. and Sharp, R. R. (2010). Bioethical and clinical dilemmas of direct-to-consumer personal genomic testing: the problem of misattributed equivalence. *Science translational medicine*, 2(17):17cm5–17cm5.

[41] Facts, O. F. (2016). Open food facts website. *Open Food Facts Website*.

[42] Fenech, M., El-Sohemy, A., Cahill, L., Ferguson, L. R., French, T.-A. C., Tai, E. S., Milner, J., Koh, W.-P., Xie, L., Zucker, M., et al. (2011). Nutrigenetics and nutrigenomics: viewpoints on the current status and applications in nutrition research and practice. *Journal of nutrigenetics and nutrigenomics*, 4(2):69–89.

[43] Frey, T., Gelhausen, M., and Saake, G. (2011). Categorization of concerns: a categorical program comprehension model. In *Proceedings of the 3rd ACM SIGPLAN workshop on Evaluation and usability of programming languages and tools*, pages 73–82. ACM.

[44] Galanakis, C. (2019). *Trends in Personalized Nutrition*. Elsevier Science.

[45] Ghodke, Y., Chopra, A., Shintre, P., Puranik, A., Joshi, K., and Patwardhan, B. (2011). Profiling single nucleotide polymorphisms (snps) across intracellular folate metabolic pathway in healthy indians. *The Indian journal of medical research*, 133(3):274.

[46] Ghoshal, S., Pasham, S., Odom, D. P., Furr, H. C., and McGrane, M. M. (2003). Vitamin a depletion is associated with low phosphoenolpyruvate carboxykinase mrna levels during late fetal development and at birth in mice. *The Journal of nutrition*, 133(7):2131–2136.

[47] Godard, B. and Hurlimann, T. (2009). Nutrigenomics for global health: ethical challenges for underserved populations. *Current Pharmacogenomics and Personalized Medicine (Formerly Current Pharmacogenomics)*, 7(3):205–214.

[48] Griffiths, A. J., Miller, J. H., Suzuki, D. T., Lewontin, R., and Gelbart, W. (2000). Genetics and the organism: Introduction. *An Introduction to Genetic Analysis*.

[49] Grocery, T. (2016). Tesco groceries website. *Tesco Grocery*.

[50] Gualtieri, J. A. and Cromp, R. F. (1999). Support vector machines for hyperspectral remote sensing classification. In *27th AIPR Workshop: Advances in Computer-Assisted Recognition*, volume 3584, pages 221–233. International Society for Optics and Photonics.

[51] Guthrie, D., Allison, B., Liu, W., Guthrie, L., and Wilks, Y. (2006). A closer look at skip-gram modelling. In *LREC*, pages 1222–1225.

[52] Hansen, M., Miron-Shatz, T., Lau, A., and Paton, C. (2014). Big data in science and healthcare: A review of recent literature and perspectives. *Yearbook of medical informatics*, 9(21):26.

[53] Hattam, V. C. (2014). *Labor visions and state power: The origins of business unionism in the United States*, volume 141. Princeton University Press.

[54] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[55] Huang, M., Cao, Y., and Dong, C. (2016). Modeling rich contexts for sentiment classification with lstm. *arXiv preprint arXiv:1605.01478*.

[56] Huang, T., Lan, L., Fang, X., An, P., Min, J., and Wang, F. (2015). Promises and challenges of big data computing in health sciences. *Big Data Research*, 2(1):2–11.

[57] Hyman, M. (2006). *Ultrametabolism: The simple plan for automatic weight loss*. Simon and Schuster.

[58] Janssens, A. C. J. and van Duijn, C. M. (2010). An epidemiological perspective on the future of direct-to-consumer personal genome testing. *Investigative genetics*, 1(1):10.

[59] Johannsen, W. (2014). The genotype conception of heredity. *International journal of epidemiology*, 43(4):989–1000.

[60] Jones, M. T. (2008). Artificial intelligence: a systems approach.

[61] Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

[62] Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

[63] Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

[64] Kasim, H., Hung, T., and Li, X. (2012). Data value chain as a service framework: For enabling data handling, data security and data analysis in the cloud. In *Parallel and Distributed Systems (ICPADS), 2012 IEEE 18th International Conference on*, pages 804–809. IEEE.

[65] Kayed, M. and Chang, C.-H. (2010). Fivatech: Page-level web data extraction from template pages. *IEEE Transactions on Knowledge and Data Engineering*, 22(2):249–263.

[66] Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[67] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[68] Kohlmeier, M. (2012). *Nutrigenetics: applying the science of personal nutrition*. Academic Press.

[69] Kosters, M. and Van der Heijden, J. (2015). From mechanism to virtue: Evaluating nudge theory. *Evaluation*, 21(3):276–291.

[70] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[71] Leonard, T. C. (2008). Richard h. thaler, cass r. sunstein, nudge: Improving decisions about health, wealth, and happiness.

[72] Leremy (2017). Supermarket market shoppers crazy rushing. *https://depositphotos.com/20530285/stock-illustration-supermarket-market-shoppers-crazy-rushing.html*.

[73] Lewontin, R. C. (1970). The units of selection. *Annual review of ecology and systematics*, 1(1):1–18.

[74] Linden, G., Smith, B., and York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80.

[75] Linko, S. (1998). Expert systems—what can they do for the food industry? *Trends in food science & technology*, 9(1):3–12.

[76] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

[77] Mariman, E. C. (2006). Nutrigenomics and nutrigenetics: the 'omics' revolution in nutritional science. *Biotechnology and applied biochemistry*, 44(3):119–128.

[78] Marti, A., Goyenechea, E., and Martínez, J. A. (2010). Nutrigenetics: a tool to provide personalized nutritional therapy to the obese. *Journal of nutrigenetics and nutrigenomics*, 3(4-6):157–169.

[79] McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D., and Barton, D. (2012). Big data. *The management revolution. Harvard Bus Rev*, 90(10):61–67.

[80] McGuire, A. L. and Burke, W. (2008). An unwelcome side effect of direct-to-consumer personal genome testing: raiding the medical commons. *Jama*, 300(22):2669–2671.

[81] McSmith, A. (2010). First obama, now cameron embraces 'nudge theory'. *The Independent*, 12.

[82] Menon, B., Harinarayan, C. V., Raj, M. N., Vemuri, S., Himabindu, G., Afsana, T., et al. (2010). Prevalence of low dietary calcium intake in patients with epilepsy: a study from south india. *Neurology India*, 58(2):209.

[83] Meyer, D., Leisch, F., and Hornik, K. (2003). The support vector machine under test. *Neurocomputing*, 55(1-2):169–186.

[84] Mihaescu, R., Van Hoek, M., Sijbrands, E. J., g Uitterlinden, A., Witteman, J. C., Hofman, A., Van Duijn, C. M., and Janssens, A. C. J. (2009). Evaluation of risk prediction updates from commercial genome-wide scans. *Genetics in Medicine*, 11(8):588–594.

[85] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[86] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[87] Miller, H. G. and Mork, P. (2013). From data to decisions: a value chain for big data. *IT Professional*, 15(1):57–59.

[88] Minton, E. A. (2013). *Belief systems, religion, and behavioral economics: Marketing in multicultural environments*. Business Expert Press.

[89] Mitchell, M., Crutchfield, J. P., Das, R., et al. (1996). Evolving cellular automata with genetic algorithms: A review of recent work. In *Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA'96)*. Moscow.

[90] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

[91] Munshi, A. and Duvvuri, V. S. (2008). Nutrigenomics: Looking to dna for nutrition advice. *CSIR*.

[92] Myerson, R. (1991). Game theory: Analysis of conflictharvard univ. *Press, Cambridge*.

[93] Neeha, V. and Kinth, P. (2013). Nutrigenomics research: a review. *Journal of food science and technology*, 50(3):415–428.

[94] Nielsen, D. E. and El-Sohemy, A. (2012). A randomized trial of genetic information for personalized nutrition. *Genes & nutrition*, 7(4):559.

[95] Nordgren, L. F., van der Pligt, J., and van Harreveld, F. (2008). The instability of health cognitions: Visceral states influence self-efficacy and related health beliefs. *Health Psychology*, 27(6):722.

[96] Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607.

[97] Osborne, M. J. and Rubinstein, A. (1994). *A course in game theory*. MIT press.

[Owen] Owen, G. Game theory. 1995.

[99] Owen, G. (1995). Game theory, 3rd.

[100] Perner, L. (2008). Food marketing consumption and manufacturing. *Avalaible from URL: http://www. consumerpsychologist. com/food_marketing. html [accessed 21 March 2011]*.

[101] Phillips, C. M. (2013). Nutrigenetics and metabolic disease: current status and implications for personalised nutrition. *Nutrients*, 5(1):32–57.

[102] Pinterest (2017). Pinterest. *https://www.pinterest.co.uk/explore/world-map-poster/*.

[103] Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

[104] Quinn, S., Bond, R., and Nugent, C. (2017). Ontological modelling and rule-based reasoning for the provision of personalized patient education. *Expert Systems*, 34(2).

[105] Raj, M., Sundaram, K., Paul, M., Deepa, A., and Kumar, R. K. (2007). Obesity in indian children: time trends and relationship with hypertension. *National Medical Journal of India*, 20(6):288.

[106] Rasmusen, E. and Blackwell, B. (1994). Games and information. *Cambridge, MA*, 15.

[107] Ravulapati, K. K., Rao, J., and Das, T. K. (2004). A reinforcement learning approach to stochastic business games. *Iie Transactions*, 36(4):373–385.

[108] Ross, S. and Savage, L. (2012). *Rethinking the Politics of Labour in Canada*. Fernwood.

[109] Rossi, P. H., Lipsey, M. W., and Freeman, H. E. (2003). *Evaluation: A systematic approach*. Sage publications.

[110] Runde, S. and Fay, A. (2011). Software support for building automation requirements engineering—an application of semantic web technologies in automation. *IEEE Transactions on Industrial Informatics*, 7(4):723–730.

[111] Saghai, Y. (2013). Salvaging the concept of nudge. *Journal of medical ethics*, 39(8):487–493.

[112] Sainath, T. N., Vinyals, O., Senior, A., and Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4580–4584. IEEE.

[113] Sak, H., Senior, A. W., and Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Interspeech*, pages 338–342.

[114] Schneider, D. (2017). Deeper and cheaper machine learning [top tech 2017]. *IEEE Spectrum*, 54(1):42–43.

[115] Scrucca, L. (2016). On some extensions to ga package: hybrid optimisation, parallelisation and islands evolution. *arXiv preprint arXiv:1605.01931*.

[116] Scrucca, L. et al. (2013). Ga: a package for genetic algorithms in r. *Journal of Statistical Software*, 53(4):1–37.

[117] Seide, F. (2017). Keynote: The computer science behind the microsoft cognitive toolkit: An open source large-scale deep learning toolkit for windows and linux. In *Code Generation and Optimization (CGO), 2017 IEEE/ACM International Symposium on*, pages xi–xi. IEEE.

[118] Sejwal, R. (2014). Game theory and human behavior. *International Research Journal of Management Sociology & Humanity*, 5:360–364.

[119] Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

[120] Smith, V. L. et al. (1992). Game theory and experimental economics: beginnings and early influences. *Toward a history of game theory*, 24:241.

[121] Sparrow, A. (2008). Speak 'nudge': The 10 key phrases from david cameron's favourite book. *The Guardian*.

[122] Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., and Demirbas, M. (2010). Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 841–842. ACM.

[123] Su, W., Wang, J., Lochovsky, F. H., and Liu, Y. (2012). Combining tag and value similarity for data extraction and alignment. *IEEE Transactions on knowledge and Data Engineering*, 24(7):1186–1200.

[124] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

[125] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT press Cambridge.

[126] Swain, S. and Sarangi, S. S. (2013). Study of various classification algorithms using data mining. *International Journal of Advanced Research in*, 2(2):110–114.

[127] Szwacka-Mokrzycka, J. (2015). Trends in consumer behaviour changes. overview of concepts. *Acta Scientiarum Polonorum. Oeconomia*, 14(3).

[128] Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

[129] Tekiner, F. and Keane, J. A. (2013). Big data framework. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 1494–1499. IEEE.

[130] Thaler, R. and Sunstein, C. (2008). Nudge: The gentle power of choice architecture. *New Haven, Conn.: Yale*.

[131] Toumazou, C., Shepherd, L. M., Reed, S. C., Chen, G. I., Patel, A., Garner, D. M., Wang, C.-J. A., Ou, C.-P., Amin-Desai, K., Athanasiou, P., et al. (2013). Simultaneous dna amplification and detection using a ph-sensing semiconductor system. *Nature methods*, 10(7):641–646.

[132] van Otterlo, M. and Wiering, M. (2012). Reinforcement learning and markov decision processes. *Reinforcement Learning*, 12:3–42.

[133] Whitley, D. and Yoo, N.-W. (1994). Modeling simple genetic algorithms for permutation problems. In *FOGA*, volume 1994, pages 163–184.

[134] Wilk, J. (1999). Mind, nature and the emerging science of change: An introduction to metamorphology. In *Metadebates on Science*, pages 71–87. Springer.

[135] Xu, D. (2014). *An integrated simulation, learning and game-theoretic framework for supply chain competition*. The University of Arizona.

[136] Yuan, G.-X., Ho, C.-H., and Lin, C.-J. (2012). Recent advances of large-scale linear classification. *Proceedings of the IEEE*, 100(9):2584–2603.

[137] Zell, A. (1994). *Simulation neuronaler netze*, volume 1. Addison-Wesley Bonn.

[138] Zhai, Y. and Liu, B. (2006). Structured data extraction from the web based on partial tree alignment. *IEEE Transactions on Knowledge and Data Engineering*, 18(12):1614–1628.

Appendix A

List of Published paper

1. Personpersonalized expert recommendation system for optimized nutrition. Chen, C.-H., Karvela, M., Sohbati, M., Shinawatra, T., and Toumazou, C. (2017). IEEE transactions on biomedical circuits and systems, 12(1):151–160. [22]

2. Chapter 12 Personalised Expert Recommendation System for Optimised Nutrition. Galanakis, C. (2019).Trends in Personalized Nutrition. Elsevier Science. [44]