

卒業論文

ウェルビーイングに有益な User eXperience を 考慮できる自立献立作成システムの開発

Development of an Automated Menu Planning Support System
that Considers User Experience Beneficial for Well-Being

富山県立大学 工学部 情報工学科

2120040 堀由隆

指導教員 奥原 浩之 教授

提出年月: 令和7年(2025年)2月

目次

図一覧	ii
表一覧	iii
記号一覧	iv
第1章 はじめに	1
§ 1.1 本研究の背景	1
§ 1.2 本研究の目的	2
§ 1.3 本論文の概要	3
第2章 UX を考慮した献立作成支援	4
§ 2.1 ウェルビーイングと UX	4
§ 2.2 多目的最適化としての献立作成	7
§ 2.3 スクレイピングによるデータ収集	10
第3章 UX が反映されるシステム	13
§ 3.1 制限食を考慮した対話型献立作成	13
§ 3.2 分析データの前処理	16
§ 3.3 ロジスティック回帰分析	18
第4章 提案手法	22
§ 4.1 UX データの収集	22
§ 4.2 UX を考慮した献立作成	25
§ 4.3 提案システムの流れ	28
第5章 数値実験並びに考察	32
§ 5.1 数値実験の概要	32
§ 5.2 実験結果と考察	35
第6章 おわりに	39
謝辞	40
参考文献	41

図一覧

2.1	パレート最適解のイメージ	8
2.2	解探索のイメージ (粒子群最適化)	8
2.3	おいしい健康のレシピページ例 [5]	11
2.4	食材価格動向調査サイトの例 [6]	12
2.5	Web スクレイピングの流れ	12
3.1	考慮するアレルギー [15]	14
3.2	ロジスティック回帰分析のイメージ	21
3.3	評価指標について	21
4.1	遺伝的アルゴリズムのイメージ [23]	26
4.2	SMOTE のイメージ	26
4.3	提案手法の流れ	29
5.1	栄養素データの例	33
5.2	最適化処理の実行画面	33
5.3	パレート解の出力	34
5.4	対話型処理による解の選択	34
5.5	ユーザ 1 のモデル評価の推移	37
5.6	ユーザ 2 のモデル評価の推移	37
5.7	ユーザ 3 のモデル評価の推移	37
5.8	ユーザ 4 のモデル評価の推移	38
5.9	ユーザ 5 のモデル評価の推移	38
5.10	ユーザ 6 のモデル評価の推移	38

表一覧

3.1	考慮する生活習慣病	15
3.2	処理したデータ一覧	19
5.1	モデル評価 (精度) の出力結果	35
5.2	精度の t 検定の結果	35
5.3	モデル評価 (適合率) の出力結果	35
5.4	適合率の t 検定の結果	35
5.5	モデル評価 (再現率) の出力結果	36
5.6	再現率の t 検定の結果	36
5.7	モデル評価 (F 値) の出力結果	36
5.8	F 値の t 検定の結果	36

記号一覧

以下に本論文において用いられる用語と記号の対応表を示す.

用語	記号
j 人目の使用者の名前	ϵ_j
j 人目の身長	α_j
j 人目の体重	β_j
j 人目の基礎代謝量 (下限)	B_j^L
j 人目基礎代謝量 (上限)	B_j^H
j 人目のアレルギー情報	x_j
j 人の有する生活習慣病	z_j
対象の日数	D
レシピの数	R
食材の数	Q
栄養素の数	N
データベース上の食材数	S
データベース上の食材番号	$d : 1, 2, 3, \dots, S$
日の番号	$k : 1, 2, 3, \dots, 3D$
栄養素の番号	$l : 1, 2, 3, \dots, N$
材料の番号	$m : 1, 2, 3, \dots, Q$
レシピの番号	$i : 1, 2, 3, \dots, R$
i 番目のレシピの名前	y_i
i 番目のレシピの献立フラグ	r_{ki}
i 番目のレシピの主菜フラグ	σ_i
i 番目のレシピの調理時間	T_i
i 番目のレシピの摂取カロリー	C_i
i 番目のレシピの調理コスト	G_i
i 番目のレシピの UX 項目の推定確率	Q_i
i 番目のレシピの m 番目の材料の名前	q_{im}
i 番目のレシピの m 番目の材料量	e_{im}
i 番目のレシピの l 番目の栄養素の名前	n_{il}
i 番目のレシピの l 番目の栄養素の量	f_{il}
d 番目の食材名	Z_d
d 番目の食材の販売単位	W_d
d 番目の食材の値段	M_d

はじめに

§ 1.1 本研究の背景

戦後の日本では急速に生活様式が欧米化し、特に食生活においてはジャンクフードやファストフードなど、手軽で高エネルギーながら栄養が偏った食品が普及するようになった。この変化により、現代では生活習慣病を患う人々が増加しており、その予防と改善が大きな課題となっている。生活習慣病は、食習慣や運動習慣、喫煙、飲酒、ストレスなど、日々の生活習慣が原因となって発症する疾患であり、脳血管疾患や心疾患といった深刻な病気を引き起こすリスクがある。特に栄養が偏った状態で続けると、生活習慣病のリスクが高まる。

生活習慣病の特徴的な問題は、その発症が徐々に進行するため、日々の生活の中で知らず知らずのうちに血管や心臓、脳にダメージが蓄積し、症状が現れるころにはすでに命に関わる疾患に進行していることが多い。このため、早期の予防が非常に重要であり、特にバランスの取れた食事の摂取が生活習慣病の予防には欠かせない要素となる。しかし、現代の忙しい生活環境において、栄養バランスを考慮した食事を毎日作るとは多くの家庭にとって大きな負担となっている。

特に、共働きの家庭や時間に追われる家庭では、食事の準備にかかる時間が限られており、手軽に食事を済ませる傾向が強くなる。結果的に、栄養が偏りがちで、外食や加工食品に頼ることが増えてしまう。これが生活習慣病を引き起こすリスクを高める要因となっている。外食や便利な加工食品が普及する一方で、家計や健康への影響を考慮すると、栄養バランスを意識した食事を効率的に作る方法が求められる。

家庭での食事作りは、栄養バランスを取るだけでなく、家族全員の嗜好や食べる時間帯、調理の手間を考慮する必要がある。特に、忙しい日々の中で毎日の献立を考えることは、時間的な余裕がない家庭では大きな負担となり、簡単に済ませてしまいがちだ。これにより、栄養が偏った食事が続き、健康への影響が懸念される。家族全員が満足できる食事を提供するためには、食事作りの効率化が求められる。さらに、食事が日々の楽しみであるべきだという点も重要である。

このような背景の中で、UX（ユーザーエクスペリエンス）を重視した献立作成システムの導入が重要となる。家庭ごとのニーズに合わせたパーソナライズされた提案を行うことができれば、栄養バランスを保ちながらも、家族全員が満足できる献立を手軽に作成することが可能になる。食材の入手しやすさ、調理の簡便さ、時間帯に適した食事提案など、家庭の実情を反映したシステムが提供されれば、献立作成の負担が軽減され、健康的な食生活が促進される。このようなシステムは、栄養士の負担を軽減すると同時に、家庭内での食事作りを効率化し、生活習慣病の予防に貢献することが期待される。

§ 1.2 本研究の目的

栄養バランスが取れた献立を作成するには、膨大なメニューの組み合わせや、それぞれの栄養素や摂取カロリーの計算など、多くの要素を考慮する必要がある。特に、忙しい日常の中で献立を考える時間がなかったり、食材や栄養を効率よく組み合わせることが難しいと感じる人々も少なくない。また、家計や時間的な制約を持つ家庭では、手軽に、かつ経済的に栄養バランスの良い食事を提供することが理想的である。しかし、献立作成における複雑な計算や調整は、時間がかかり面倒に感じられることが多い。多忙な現代人にとって、献立作成に要する労力や時間は大きな障壁となっており、特に家庭での食事作りが負担となることが多い。

加えて、学校給食や病院食など、公共機関での献立作成を担う栄養士には、毎日の食事計画において、栄養計算や食材費用の計算、さらには食材の調達や調理の簡便さを考慮しながら、何度も献立を見直す作業が求められている。このプロセスは非常に時間を要し、繰り返し行う作業であるため、その負担は大きい。これにより、献立作成業務は非常に負担が大きく、特に多忙な現場ではその負荷が問題となっている [3]。病院などでは、食事が患者の日常生活の中心であり、食事が患者の楽しみの一部であるため、食事の内容や栄養だけでなく、患者の好みに合った献立を提供することが求められる。このような現場では、食に対する専門性を高めるために日々研究や開発が行われているが、それに伴い献立作成の業務負担も増大している。患者一人一人のニーズに応じた献立作成が求められる一方で、そのための業務負担の軽減も重要な課題となっている。

このような背景を踏まえ、本研究では、献立作成の負担を軽減し、効率的に栄養バランスを取った食事を提供するシステムを提案する。具体的には、膨大な料理データを基に、調理時間や食材コスト、さらには個々の身体的な情報や嗜好に基づいた最適な献立を自動的に提案することを目指す。このアプローチにより、家庭や病院などの現場で、手間をかけずに効率的にバランスの取れた食事を提供できるようになる。献立作成のために収集するデータには、Web上の複数のレシピサイトからスクレイピングによって集めた料理データを活用する。具体的には、各料理に必要な食材やその栄養価、カロリーなどの情報に加え、食材ごとの価格データも蓄積する。これにより、食材のコストパフォーマンスを考慮した献立提案が可能となり、家計にも優しい食事の提供が実現できる。

また、組み合わせ多目的最適化問題として献立を作成する際、遺伝的アルゴリズムを応用した非優越ソート遺伝的アルゴリズム (NSGA-II) を使用し、最適な献立を提案する。これにより、栄養バランスを保ちながらも、個々のニーズに最適な献立が提供される。また、ロジスティック回帰分析を用いて、ユーザーが過去に評価した献立に基づく UX 項目（例：調理のしやすさ、食材の入手しやすさ、時間帯に適した献立かどうかなど）を予測し、その結果を次の献立提案に反映させる。このアプローチにより、ユーザーごとにパーソナライズされた献立推薦が可能となり、効率的かつ個別の好みに合った食事を提供することができる。

最終的には、栄養バランスの取れた食事の提供がより効率的かつ満足度の高いものとなり、ユーザーのニーズに最適化された献立作成が実現される。このシステムにより、献立作成にかかる時間や手間を軽減し、家庭内での食事作りを効率化することで、生活習慣病の予防や健康的な食生活の促進に貢献することが期待される。

§ 1.3 本論文の概要

本論文は次のように構成される。

第1章 本研究の背景と目的について説明する。背景では栄養バランスの摂れた献立を作成することの難しさと、自動で献立を作成することの重要性について示す。目的は制限食を考慮した多目的遺伝的アルゴリズムによる最適な自動献立作成について提案することを述べる。

第2章 多目的最適化による自動献立作成システムの概要と、ウェルビーイングとUXについて説明する。

第3章 UX項目を考慮した献立作成の概要と、ロジスティック回帰分析の仕組みを説明する。また、本研究で用いる制限食及びブラウザベースのシステムについて説明する。

第4章 提案手法の中でUX項目データの収集部分と、NSGA-IIによる多目的最適化によって最適な献立を対話型で出力する部分について説明する。
その後、提案手法について説明する。

第5章 提案手法に基づいて自動献立作成システムを構築して、実際に献立の作成を行った結果を示す。そして、本研究の提案手法によって得られた結果が有意であることを示す。

第6章 本研究で述べている提案手法をまとめて説明する。また、今後の課題について述べる。

UXを考慮した献立作成支援

§ 2.1 ウェルビーイングとUX

ウェルビーイング（well-being）とは、個人の健康や幸福、生活全体の質を示す包括的な概念であり、身体的、精神的、社会的、経済的な側面が相互に関連する多次元的な状態を指す。一般的に、ウェルビーイングは単なる健康状態にとどまらず、個人が自らの生活に満足しているか、心理的に充実しているか、社会的に適応しているか、そして経済的に安定しているかをも含む広範な指標である。

ウェルビーイングの主要な構成要素としては、以下のものが挙げられる：

- **身体的ウェルビーイング**：健康的な生活習慣、適切な栄養摂取、運動習慣、十分な睡眠など、身体的健康を維持することが基本である。
- **精神的ウェルビーイング**：ストレスの管理、感情的安定、ポジティブな感情や自己肯定感、心理的回復力など、精神面の健康が重要な要素となる。
- **社会的ウェルビーイング**：他者との良好な関係、社会的支援、所属感やコミュニティとのつながりなど、社会的なつながりが個人のウェルビーイングに大きな影響を与える。
- **経済的ウェルビーイング**：経済的な安定性、収入の水準、労働の充実感など、経済的な側面もウェルビーイングに大きな役割を果たす。
- **自己実現および精神的充足感**：個人が自己の価値観や目標に沿った生活を送り、自己実現を感じることが精神的ウェルビーイングを高める。

このように、ウェルビーイングは多面的であり、単一の健康状態や幸福感に限定されるものではなく、個人が社会全体の中でどのように生き、どのように自分自身を実現していくかというプロセスを包括的に反映する概念である。また、ウェルビーイングは社会政策や公共の福祉、企業戦略など、さまざまな分野においても重要な指標となっており、国際的には国民の幸福度や生活満足度を測るための指標としても利用されている。

ユーザーエクスペリエンス（UX）とは、ユーザーが製品やサービスを使用する過程で得る経験全体を指す概念であり、使いやすさ、効率、満足度などが重要な要素となる。UXは単に製品の操作性だけでなく、ユーザーがその製品を使うことに対してどれだけ満足しているか、どれだけ快適に感じるか、どれだけストレスなく利用できるかを反映する広範な指標である。

UXの主要な構成要素としては以下のものが挙げられる：

- **使いやすさ:** 製品やサービスの直感的な操作性や、ユーザーが迷うことなく目的を達成できるかどうかが重要である。
- **効率性:** ユーザーがタスクをどれだけ迅速かつ簡単に完了できるか。操作に必要な時間やステップ数が少ないことが求められる。
- **満足度:** ユーザーが製品を使用することで得られる満足感や喜び。良好な UX はユーザーが製品を使用し続ける動機となる。
- **信頼性:** 製品やサービスが安定して動作し、ユーザーが安心して使用できることが求められる。
- **感情的反応:** ユーザーが製品やサービスを使用した際に、どのような感情を抱くか。使い心地やデザインなどが感情的な反応を引き起こす。

UX は、製品やサービスがユーザーにどれだけ適応しているか、どれだけ効率的に満足度を高められるかという観点から、サービスデザインやユーザビリティテスト、フィードバックなどを通じて向上させていくプロセスである。

ウェルビーイング、UX、献立作成プログラムへの関連: ウェルビーイングの考え方は、ユーザの身体的、精神的、社会的、経済的な状態を考慮した献立作成プログラムにおいて非常に重要である。プログラムでは、各ユーザの健康状態やアレルギー情報、生活習慣、精神的な充足感を反映させた献立の提案を行うことが求められる。身体的ウェルビーイングに基づく栄養素のバランスを考慮した献立、ストレスや心理的安定をサポートする食材の提案、社会的なつながりを促進するような共食を意識したメニュー作成が必要であり、これらはユーザの全体的なウェルビーイングを向上させる手段となる。

UX は、ユーザーが献立作成プログラムを使う際の体験にも深く関わる。ユーザーが自分のニーズに合わせた食事提案を簡単に得られ、心地よく操作できることが重要である。身体的健康をサポートする栄養バランスを考慮しつつ、ユーザーの好みや生活習慣にマッチした提案を行うことで、満足度が高まり、継続的な使用が促進される。また、食事の提案を通じて、ユーザーが自己管理を感じられるような UI デザインを取り入れることで、感情的な充足感を得ることも可能である。UX を重視した献立作成プログラムは、ユーザーのウェルビーイングを向上させ、健康的な生活の支援をする重要なツールとなる。

ウェルビーイングを高める UX デザインの例

マインドフルネスアプリ

ユーザーインターフェース (UI) の配慮: Headspace や Calm のようなアプリでは、視覚的に落ち着けるデザインが使われている。シンプルなレイアウトと柔らかな色調が、ユーザーにリラックス感を与え、マインドフルネスを実践しやすくしている。

パーソナライズ機能: ユーザーの進捗に合わせて瞑想セッションをカスタマイズする機能があり、ストレスレベルに応じたコンテンツの提供や、習慣化を促すリマインダー機能が役立っている。

フィットネスアプリ

モチベーションを高める要素: Nike Training Club や MyFitnessPal では、ユーザーの

運動記録や目標達成度をグラフやダッシュボードで可視化し、達成感を感じやすくしている。また、定期的にモチベーションを維持できる通知が届く仕組みがある。

ソーシャル機能: ユーザーが他のユーザーと成果を共有したり、チャレンジを通じて競い合ったりする機能が提供されており、健康維持への意欲を高める。

ソーシャルメディアの使用制限機能

利用時間のトラッキング: Instagram や Facebook などでは、1日の使用時間をトラッキングし、制限時間を超えると通知が表示される。これにより、過剰な使用を防ぐことができる。

健康的な休憩の提案: 長時間使用後に「休憩を取る」などのリマインダーが表示され、ユーザーが自分の使用状況を意識できるようになっている。

E コマースのストレス軽減機能

簡潔で透明な情報提供: Amazon のようなプラットフォームは、商品の詳細や料金、返品ポリシーを明確に表示しており、購入時の不安を減らしている。

フォローアップ機能: 購入後、配送状況の通知や、満足度調査が送られることで、ユーザーは安心感を得られる。

高齢者向けサービス

視覚・聴覚への配慮: 高齢者向けのアプリやウェブサイトでは、大きなフォントや音声ナビゲーションが使われており、視覚や聴覚に障害があるユーザーも使いやすくなっている。

簡易化されたインターフェース: 複雑な操作を避け、直感的に使えるインターフェースが提供されている。例えば、ボタンサイズや配置を工夫し、高齢者でも簡単に操作できるようになっている。

障害者支援ツール

スクリーンリーダー対応: 視覚障害者向けに、スクリーンリーダーが動作するアプリやサイトが増えており、テキストの読み上げ機能が利用されている。

カスタマイズ可能なインターフェース: 身体的制約があるユーザー向けに、文字サイズやインターフェースの色、操作方法をカスタマイズできるツールが提供されている。

ゲームとウェルビーイング

健康管理を取り入れたゲーム: フィットネスゲームの進化: ポケモン GO のように、ゲームを通じて歩いたり運動を促進したりすることができ、ユーザーは楽しみながら健康維持できる。ゲーム内の達成感や報酬が、継続的なモチベーションを提供している。

健康データの活用: ヘルスケアアプリとの連携で、ゲーム内での歩数や運動量が記録され、ユーザーは自分の健康状態を確認できる。

§ 2.2 多目的最適化としての献立作成

多目的最適化とは、「制約条件のもと、複数の選択肢を組み合わせて何か結果を出すとき、その結果（目的関数）を最小、もしくは最大にすること」である。多目的最適化の利点として自動化による結果が出るまでの作業時間が削減されることや、答えを導くのに現実的ではない時間がかかる問題を解くことができることがある。

最適化問題の種類の一つとして、組み合わせ最適化問題が挙げられる。本研究の自動献立作成システムはこれに分類される。組み合わせ最適化問題とは、様々な制約のもとで多くの選択肢の中から、ある評価（価値）を最もよくする変数の値（組み合わせ）を求めることである。

献立における制約条件として、何日分の献立を作成するか、カロリーをどのくらい制限するか、特定の栄養素を最低でもどのくらい取得するか、などが挙げられる。また、目的関数として、調理時間の最小化や個人の嗜好の最大化、材料コストの最小化などが挙げられる。

しかし、組み合わせ最適化を解く場合、目的関数がトレードオフになる関係がある場合がある。トレードオフとは、何かを得ると別の何かを失う相容れない関係のことである。食事を例に挙げるとすると、一般的に高カロリーな食生活によって食の満足度は上がるが、体重が増えて栄養に支障が生じる。逆に健康を意識してダイエットを行えば、食の満足感が減る。この場合、「高カロリーな食生活」と「健康」の関係性がトレードオフの関係になっている。

目的関数がトレードオフの関係である場合、一方の目的関数の最小化あるいは最大化が、他方の目的関数の最小または最大化に悪い影響を及ぼすため、単一目的の最適化問題とは異なり、複数の目的関数をすべて満たすような一つの最適解を得ることは困難である。多目的最適化での探索では、パレート最適解と呼ばれる概念を導入する必要がある。

パレート最適解とは、ある目的関数を満たそうとしたときに他の目的関数が犠牲になり満たされなくなってしまう解のことであり、非劣解とも呼ばれる。反対に、パレート最適ではないような解のことは劣解と呼ばれている。

また、パレート最適解は一般的には1つとなることはほとんどなく複数となる場合がほとんどであるため、集合となる。パレート最適解の集合のイメージを図3.2に示す。複数のパレート最適解を、目的関数空間に添付したときに形成される曲線は、パレート最適フロントと呼ばれる。実際にはこのパレート最適フロントの中から解を選択することになるのである。

また、一般的に最適化問題には、実行可能領域という、制約条件を満たす領域の内側に複数の局所的最適解を持つ。局所的最適解とは、その近辺では最も良い解であるが、実行可能領域全体で考えたときに、最も良い解になるとは限らない解のことである。局所最適解に対して大域的最適解とは、制約条件を満たしなおかつ実行可能領域全体で、最も良い解であることが保証されているものとなる。

大域的最適化解は、通常は単に最適解と呼ばれることが多い。しかし目的関数が凸関数で制約集合が凸集合である非凸計画問題や、組合せ最適化問題などにおいては、局所的最適解との区別を強調したい場合に、大域的という形容詞をつけて大域的最適解と呼ばれる場合が多い。

局所解を回避する方法のうち、例として遺伝的アルゴリズムを用いて最適化を解くとい

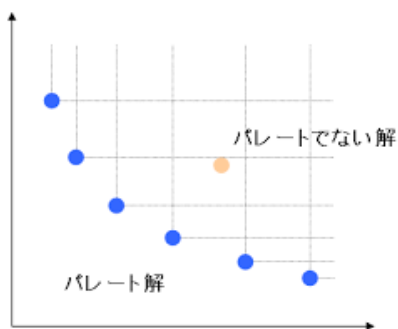


図 2.1: パレート最適解のイメージ

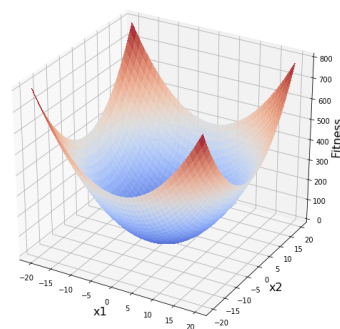


図 2.2: 解探索のイメージ (粒子群最適化)

う場合、突然変異率のパラメータを変更する方法がある。突然変異率は、低すぎると局所解に陥りやすくなる。しかし高すぎるとランダム探索になってしまうため、調整が必要となるパラメータとなる場合が多い。その他には、最適化アルゴリズムの1つである確率的勾配降下法は、確率的に局所解を抜け出せる可能性があるとされている。ここで、多目的最適化問題の定式化を行う。

多目的最適化問題の定式化

多目的最適化問題は、 $x = (x_1, x_2, \dots, x_N)$, $f_i(x)$, $i = 1, 2, \dots, n$ を目的関数として、 $g_k(x)$, $k = 1, 2, \dots, m$ を制約条件式とすると、

$$\underset{x}{\text{minimize}} \quad \{f_1(x), f_2(x), \dots, f_n(x)\} \quad (2.1)$$

$$\text{subject to} \quad g_k(x) \leq 0 \quad k = 1, 2, \dots, m \quad (2.2)$$

のように定式することができる。式 (3.1) で与えられるベクトル関数を式 (3.2) の制約条件を満たした状態で最適化する問題を多目的最適化問題と呼ぶ。

次に、最適化問題を解く様々な手法について説明する。

差分進化法

差分進化法とは、進化的アルゴリズムの一種であり、確率的な直接探索によって解集団を用いた多点探索を行うアルゴリズムである。差分進化法は非線形問題、微分不可能な問題などの、現実的な実行時間では厳密な解を導くことが困難な問題に対して近似解を求めることが可能であり、様々な最適化問題に適応されることが多い。差分進化法のアルゴリズムは、以下のステップで構成される。

1. 初期集団の生成：解候補となるベクトル集団をランダムに生成。
2. 変異操作：集団内のベクトルの差分を利用して新しい候補ベクトルを生成。
3. 交叉操作：既存のベクトルと変異後のベクトルを組み合わせ、新しい候補を作成。
4. 選択操作：目的関数の評価に基づいて、次世代の集団を決定。

このプロセスを終了条件（例：世代数、収束基準）が満たされるまで繰り返す。差分進化は非線形、非微分可能な関数に対しても適用可能であり、現実の多くの問題で効果を発揮する。

Nelder-Mead 法

$n + 1$ 個の頂点からなる、 n 次元ユークリッド空間の実数値関数を $f(x)$ を与え、そのもとで $f(x)$ の最小値を微分に頼らずに求める方法の一つである。設計変数が 2 つの場合、2 次元平面上の 3 つの点を初期値として与え、各ベクトルの目的関数を計算し、中でも最大値をとる点を、鏡像、拡大、収縮、縮小の 4 つの操作どれかを使い移動する。この操作を繰り返し行うことにより、全ての点を最小値に近づけていく、というアルゴリズムになっており、滑降シンプレックス法やアメーバ法とも呼ばれている。アルゴリズム内の操作を説明すると以下のようになる。

鏡像：最大値を持つ頂点を中心に反射。

拡大：反射後の点を拡張方向にさらに移動。

収縮：最大値の点を収束方向へ移動。

縮小：すべての頂点を中心点に向かって縮小。

これらの操作を繰り返すことで、シンプレックス全体が目的関数の最小値に近づいていく。Nelder-Mead 法は計算コストが低く、実装も簡単なため、低次元かつ連続変数の最適化に広く用いられる。

多目的粒子群最適化

多目的粒子群最適化 (Multiobjective Particle Swarm Optimization: MOPSO) とは、粒子群最適化の手法を用いて多目的最適化を解く方法である [2]。

PSO は、探索空間内において多くの粒子を用いて探索を行う、群知能の一種であり、魚群や鳥の群れにおいて、1 匹が食料を発見できたり、安全であったりといった意味で、良さそうな経路を発見すると、群れの残りは素早くそれに倣うという特徴を応用したアルゴリズムである。粒子群最適化によって解の探索を行っている様子を図 3.3 に示す。

MOPSO の基本的なアルゴリズムは、PSO と同様で、パレート解は無数に存在し、今までの単目的のように最良な解が明らかではないため、粒子の最良解である personal best、粒子群全体での最良解である global best をどのように選択するかが問題となる。

また、gbest はパレート解の中から、なるべく多様な解を求めるために、解空間での密度を考慮し、密度の薄い所から重点的にサンプリングする方法が存在する。

多目的進化アルゴリズム

多目的進化アルゴリズムは 2007 年に提案された [3] アルゴリズムのことであり、多目的最適化問題を、スカラー化する関数によって、複数の単目的問題に分割して解く手法のことである。スカラー化する関数には、Weighted Sum や Tchebycheff, Achievement Scalarizing Function などがあり [4]、中でも Weighted Sum は最も単純で、各目的関数に重み付けをした値を合計し、スカラー化した関数値とするものとなっている。多目的進化アルゴリズムは初期化、交叉、突然変異、評価点の更新、解の更新といった流れを、終了条件を満たすまで繰り返すというアルゴリズムになっている。

§ 2.3 スクレイピングによるデータ収集

現在、cookpad やクラシル、おいしい健康、ボブとアンジー [5] などの料理レシピサイトと呼ばれるサイトが多数存在する。これらのサイトには、料理名、料理の画像、和食や洋食、主菜や副菜などの料理のジャンル、麺類や丼もの、鍋料理などの料理タイプ、料理につかう材料の名前とその材料の数、調理の工程、摂取カロリー、調理時間、得られるすべての栄養素などの情報がレシピサイトに掲載されている。

レシピサイトの1つであるおいしい健康の料理レシピ名、レシピの画像が乗っているページの例を図 2.3 に示す。また、生鮮食品や加工食品、畜産品などの最低、平均、最高販売価格の価格動向を先月や前年同月と比較している情報を提供している Web サイトも存在している (図 2.4 参照)。

本研究では、献立作成システムにおいて摂取栄養量をみたすかどうかという制約条件のもと献立を作成するため、料理から摂取できる栄養量をできるだけ細かく掲載されている複数の料理レシピサイトからレシピデータを取得する。また、食品価格動向を調査している Web サイトである小売物価統計調査による価格推移というサイト [6] から、食材とその価格のデータを取得する。また、これらのデータはスクレイピングという手法で取得する。

スクレイピング

スクレイピングとは、データを収集し、かつ目的に合わせて加工することである。特に、Web 上から必要なデータを取得することを、Web スクレイピングと呼ばれている。Web スクレイピングの流れについて図 2.5 に示す。様々なツールやプログラミングでスクレイピングを自動化することで、Web データの収集にかかる手間や時間は大幅に削減が可能である。

スクレイピングと似ている意味の言葉にクローリングがある。クローリングとは、Web 状で様々なサイトを巡回し、情報の保存や複製など様々なことを行うことを指す。クローリングとスクレイピングはともに情報を収集手段ではあるが、クローリングが巡回に焦点を当てている一方でスクレイピングは情報の抽出に焦点を当てている。

また、企業や公共機関は、情報やデータを提供してくれることもあり、その際に使われている仕組みは API と呼ばれている。クローリングやスクレイピングをする前に、必要な情報が API によって提供されているかどうかまず確認することが大切になる。

Web スクレイピングに主に用いられるツールとして、BeautifulSoup4 や、Selenium がある。ログインやボタンのクリックなどの、マウス操作が必要な Web サイトや、JavaScript で記述されている Web ページのスクレイピングするときは Selenium が用いられている、それらの処理を必要としない Web サイトには、高速でスクレイピングができる BeautifulSoup4 が使用されることが多い。

Beautiful Soup4

BeautifulSoup4 とは、Web サイト上の HTML から、必要なデータを抽出するための Python のライブラリである。Beautifulsoup4 でスクレイピングする際、最初に対象の Web ページから HTML を取得する必要がある。

HTML を取得する方法として、同じく Python のライブラリである、Requests の get 関数や、Selenium の page_source 関数を使うなどの方法がある。上記の方法によって取得された HTML テキストを、BeautifulSoup4 の BeautifulSoup 関数



(a) 料理名とイメージ

食塩相当量	0.9 g	マグネシウム	24 mg	ビタミンB12	0.3 µg
たんぱく質	16.8 g	リン	177 mg	葉酸	28 µg
脂質	14.2 g	鉄	0.7 mg	ビタミンA	44 µg
炭水化物	3.5 g	亜鉛	1.6 mg	ビタミンD	0.4 µg
糖質	3.2 g	ヨウ素	0 µg	ビタミンK	35 µg
食物繊維	0.3 g	コレステロール	89 mg	ビタミンE	0.8 mg
水溶性食物繊維	0.0 g	ビタミンB1	0.11 mg	飽和脂肪酸	4.38 g
不溶性食物繊維	0.3 g	ビタミンB2	0.16 mg	一価不飽和脂肪酸	6.71 g
カリウム	343 mg	ビタミンC	4 mg	多価不飽和脂肪酸	1.86 g

(b) 得られる栄養素量と必要食材料

図 2.3: おいしい健康のレシピページ例 [5]

に渡すことで、BeautifulSoup オブジェクトを作成することができる。また、そのオブジェクトから class を検索することで Web サイトの必要な情報を抽出する。

class を検索するとき、条件を満たすひとつの要素を取得する select_one 関数や、条件に合う条件のすべてを取得する select 関数、find 関数などがある。select と find の違いは引数を指定する条件の指定方法がある。前者は、CSS セレクタを指定して要素を取得し、後者は class 名や属性キーワードを指定して検索し、class を取得する。これらの関数から取得した Tag オブジェクトである要素から、内部テキストのみを取得するためには、get_text 関数を使用することで取得することができる。

Selenium

Selenium は、Web ブラウザの操作を、自動的に操作することを可能にするライブラリである。元々は、Web アプリケーションの UI テストだったり、JavaScript のテストをする目的などで開発されていたが、テスト以外にも、Web サイトのクロールや、タスクの自動化など、多岐にわたる用途で利用されている。

スクレイピングしたレシピ情報は、1つの料理につき1つの CSV ファイルで出力され、保存される。また、食材と価格データは1つの CSV ファイルにすべて出力する。それらの CSV ファイルを1つのデータベースに蓄積し、本研究で使用する自動献立作成システムの入力データとして利用する。

Web サイトからテキストをスクレイピングする際には、Python で記述したプログラムを使用する。まず、Python のライブラリである urllib を使って、目的の Web ページの URL を渡すことで、アクセスした際の HTML データを抽出する。次に、HTML や XML を解析することができる Python のライブラリの1つである BeautifulSoup4 を用いて Web ページ内の必要な要素を取得する。

上図のレシピデータ例に含まれている材料名を、食材のデータの材料名と照会し、その材料の必要な量と販売単位、販売価格から、各材料にかかる費用を全て計算し、料理にかかるコストを各レシピごとに計算する。レシピデータに含まれる材料名と食材価格データに含まれる材料名を照らし合わせる際に、微妙に違いが発生することがある場合、2つの材料名の文字列がどれほど一致しているかという類似度を計算し、類似度がしきい値よりも



図 2.4: 食材価格動向調査サイトの例 [6]



図 2.5: Web スクレイピングの流れ

大きい場合に一致しているとしてコストの計算を行う。しきい値は、類似度計算に用いた Python のライブラリ関数にて、デフォルトの値である 0.65 を用いる。

文字の類似度を測定する際には、Python に標準で搭載されているライブラリである `diffib` を利用する。類似度計算をし、一致するものが見つからなかった場合は、ショッピングサイトである楽天市場の食品のカテゴリから、その材料名で検索をする。その後材料名とその材料の価格あたりの量をスクレイピングして食材価格データベースに追加する。ここで、類似度計算でしようした `diffib` とその技術について説明する。

diffib

`diffib` は文字比較を行うために使う python 標準モジュールである。 `diffib` は、2つの文字列の類似度を表示する `SequenceMatcher` クラスや、リストからキーワードに類似した文字列を抽出する `get_close_matches` 関数などの機能がある。 `SequenceMatcher` クラスは、文字列同士の連続する共通部分を抜き取り、その抜き出した文字列の前後に対しても同様の処理を繰り返す、ゲシュタルトパターンマッチングというアルゴリズムを使用して、文字列の類似度計算処理とその表示を行う。 `get_close_matches` 関数は、特定のキーワードに類似した文字列を取得するために、マッチさせたい文字列と、マッチさせる文字列のリストを指定するほかに、マッチされた文字列のうち、上位の何件までを返すのか、何%以上の一致率ならば表示をするかなどの指定も可能となっている。

ゲシュタルトパターンマッチング

`diffib` で用いられている技術であるゲシュタルトパターンマッチングは、2つの文字列の類似度を判定するために用いられるアルゴリズムである。このアルゴリズムは Ratcliff, Obershelp によって 1983 年に考案された [?]。このアルゴリズムは、Ratcliff/Obershelp Pattern Recognition と呼ばれることもある。2つの文字列 S_1 , S_2 の類似度 D_{ro} は、

$$D_{ro}(S_1, S_2) = \frac{2K_m}{|S_1| + |S_2|} \quad (2.3)$$

で表すことができる。ここで、 K_m はマッチした文字の数であり、 D_{ro} は 0 から 1 の範囲となり、1 に近いほど類似度が高く、0 に近いほど類似度が低くなっている。

UXが反映されるシステム

§ 3.1 制限食を考慮した対話型献立作成

対話型システムとは、ユーザーとシステムが相互に情報をやり取りする形式のシステムで、ユーザーがシステムに対して指示を出し、システムがその指示に応じた応答を返す仕組みを指す。これにより、ユーザーはシステムと対話をする感覚で操作を行い、システムはその情報に基づいて処理を進める。本論文の献立作成システムでは、対話型システムを活用して、ユーザーがアレルギーや制限食を考慮した献立を作成できるようにしている。ユーザーがインターフェースを通じて、アレルギー情報や食事制限に関する詳細を入力すると、システムはその条件を踏まえた献立を提案する。また、スライダーを使用して、二つの変数に基づくパレート解を調整し、順位付けされた献立を表示することも可能だ。これにより、ユーザーはより直感的かつ柔軟に自分のニーズに合った献立を作成することができる。また献立を作成するにあたって、人によってはアレルギーを含む食品や生活習慣病による制限食を考慮しなければならない。制限食とは、個人の健康状態、病気の状態に合わせてカロリーや塩分などを制限する食事のことである。身体の状態に応じてある程度の制限を加えた食事療法は、間接的な疾病の改善や病気を悪化させないための重要な役割をはたしている。食事療法には、生活習慣病をはじめとする病気の予防したり、健康診断で「要注意」と診断された場合に状態を改善することができたり、すでに病気と判断された場合に病気を悪化させないなどの効果がある。制限食には様々な種類があり、病態にあったものを選択する必要がある。

食事療法は生活習慣病治療の基本であり、合併症やさらなる悪化を防ぐには正しい食事療法を毎日続ける必要がある。また、食事療法による制限食は、患っている病気によって制限する栄養素が異なる場合がある。次に、食事療法によって予防や改善ができる主な生活習慣病を紹介する。

糖尿病

糖尿病とは、インスリンというホルモンの不足や作用低下が原因で高血糖状態が続く病気である。糖尿病は1型と2型に分けられており、1型糖尿病は、主に自己免疫によって膵β細胞の破壊を生じ、インスリンの欠乏を来して発症する糖尿病である。2型糖尿病はインスリン分泌量低下を来す複数の遺伝因子に、過食、運動不足などの生活習慣に起因する内臓脂肪型肥満が加わり発症する糖尿病である。

糖尿病を改善するために特に調整しなければならない栄養素は炭水化物、脂質、食物繊維である。炭水化物においては、1日あたりの炭水化物摂取量を100g以下とす



図 3.1: 考慮するアレルギー [15]

る炭水化物制限が、肥満の是正に有効だとする研究結果から、糖尿病治療における炭水化物制限の有効性が注目されている [10]。また、脂質においては、日本糖尿病学会によると脂質の摂取量を必要推定エネルギーの 15~25% に抑えることが推奨されている。また、食物繊維において、糖尿病の発症リスクとの定量的解析を試みたメタ・アナリシスでは、食物繊維の平均摂取量は 20g/日を超えた時点から有意な低下傾向が見られている。

腎臓病

腎臓病とは、腎臓の糸球体や尿細管が冒されることで腎臓の働きが悪くなる病気のことである。腎臓の機能は一度失われると、回復することがない場合が多く、慢性腎不全と言われる病態になることがある。腎機能障害が進行してきた場合には、たんぱく質制限、塩分制限、カリウム制限などの食事療法を行うことにより、腎機能障害の進行を抑え、慢性腎臓病の合併症を予防することができる。たんぱく質の摂取量を制限することによって、腎機能低下の原因の一つである尿たんぱく、高リン血症の発生を軽減することができる [11]。日本腎臓学会のガイドラインでは、たんぱく質制限を行う場合は、1日のたんぱく質の摂取量を標準体重当たり 0.6~0.7g とすることが推奨されている [12]。

また、塩分の摂取量を制限することにより血圧が低下し、末期腎不全に陥るリスクが低くなることがわかっている。日本腎臓学会のガイドラインでは、腎臓病患者の食塩摂取量として、1日の摂取量が 3g 以上、6g 以下が推奨されている [12]。また、腎機能が低下すると、体内のカリウムの排泄も低下し、「高カリウム血症」を患う可能性がある。したがって、カリウム制限が必要となる。血清カリウム値 5.5mEq/L 以下を目標に 1日カリウム摂取量を 1500mg 以下に制限する必要がある。

脂質異常症

脂質異常症とは、血液中の脂質の値が基準値から外れた状態のことをいう。血液中の LDL コレステロール（悪玉コレステロール）、HDL コレステロール（善玉コレステロール）、トリグリセライド（中性脂肪）の値のいずれかが異常値であれば、脂質異常症と診断される。脂質異常症は、動脈硬化生疾患、特に心筋梗塞及び脳梗塞の危険

表 3.1: 考慮する生活習慣病

	糖尿病	腎臓病	脂質異常症	高血圧
たんぱく質(g)	健常者と同じ	標準体重当たり 0.6~0.7g	健常者と同じ	健常者と同じ
脂質(g)	総エネルギーの 15~25%	健常者と同じ	総エネルギーの 15%以下	健常者と同じ
炭水化物(g)	100g/日以下	健常者と同じ	健常者と同じ	健常者と同じ
塩分(g)	設定なし	3.0g/日以上6.0g/日以下	設定なし	6.0g/日未満
食物繊維(g)	20.0g/日以上	設定なし	20.0g/日以上	20.0g/日以上
カリウム(mg)	設定なし	1500mg/日以下	設定なし	3510mg/日以上
コレステロール(mg)	設定なし	設定なし	200mg/日以下	設定なし

因子となる疾患である。コレステロール過剰に摂取すると血中のコレステロール値が上昇し、脂質異常症が重症化してしまうため、コレステロールの摂取量を調整する必要がある。

また、日本動脈硬化学会による「動脈硬化性膝下に予防ガイドライン 2017 年版」では、1 日のコレステロールの摂取量を 200mg とすることにより、コレステロール低下し、脂質異常症の重症化を防ぐことが期待できるとしている。ほかにも、厚生労働省によると、1 日の食物繊維の摂取量を 20g 以上にすることや 1 日当たりの脂質の摂取量を総エネルギーの 15%以下にする推奨している [13]。

高血圧

高血圧とは、収縮期血圧及び拡張期血圧のいずれかが基準値を超えて上昇した状態で、診察室血圧では 140/90mmHg 以上と定義されている。高血圧が続いていて動脈硬化が進むと、動脈硬化が起こった部位ごとに様々な症状が現れる。

高血圧の要因として、塩分を過剰に摂取することによる血圧上昇が大きな要因となるため、塩分制限が必要となっている。日本高血圧学会による「高血圧治療ガイドライン 2019」によると、高血圧者の減塩目標を食塩 6g/日未満としている。また、カリウム摂取量増加によって高血圧者にとって血圧低下効果を認めた。厚生労働省によると、カリウムの摂取量を 3510mg 以上摂取することが推奨されている。さらに、1 日の食物繊維の摂取量を 20g 以上にすること推奨されている [14]。

食物アレルギーを有する人には、症状が出ないように原因となる食品を除去する「除去療法」がある。除去療法とは、原因となる食べ物を除去することであり、例えば、卵アレルギーの場合は卵が含まれる食品を摂取しないなど除去をするなどがあげられる。また、除去療法をする場合、特に複数の食物アレルギーがある場合には栄養バランスが取れなくなる場合があるので注意する必要がある。

本研究は健常者のほかに、制限食を必要とする生活習慣病を患った人、生活習慣病を予防したい人、アレルギーを持っている人でもバランスが取れた献立が出力されるような献立作成システムを作成することを目的とする。本研究で考慮する生活習慣病の数値をまとめたものを表 3.1 に示す。対象となる献立作成システムは糖尿病、高血圧、脂質異常症、腎臓病とする。また、アレルギーの対象項目として、重い症状を引き起こしやすい、あるいは症例数が多く、「特定原材料」として表示義務がされている 7 品目と、「特定原材料」に準ずるものとして表示が推奨されている 21 品目の計 28 品目とする。

§ 3.2 分析データの前処理

ロジスティック回帰分析を行うために必要なデータの前処理について、以下の作業を行った。これらの手順は、スクレイピングした献立の情報をもとに、モデルが適切に学習できるようにデータを整形するための重要なステップだ。

食材の有無の処理

最初に行ったのは、各献立に含まれる食材の有無を特徴量として取り扱うことだ。献立情報をスクレイピングし、それぞれの献立に登場する食材があるかどうかを確認した。例えば、「鶏肉」「玉ねぎ」「キャベツ」などの食材を列挙し、それぞれがその献立に含まれているかどうかを二値（1 または 0）で表現した。

この処理により、各食材がその献立に存在するかないかを明示的に特徴量として扱えるようになり、モデルが食材の有無が結果にどう影響するかを学習することができる。また、食材の選定には「ミリアンストア」[16]、「エバラ食品」[17]を参考にして行い、40種類の食材が選ばれている。

調理工程から調理動作のカウント

次に、献立に含まれる調理工程をもとに調理動作の回数をカウントした。調理工程に記載されている動作（例：「炒める」「煮る」「焼く」など）をリストアップし、それぞれの献立がどれくらいの回数でその動作を行うのかを数えた。

この特徴量を作成することで、献立がどれほど複雑か、または調理にどれくらいの手間がかかるかといった情報を反映させることができる。調理動作の数が多ければ、一般的に調理時間が長くなる可能性があるため、時間やカロリーといった他の特徴量と組み合わせることで分析することができる。

正規表現を利用した調理動作の抽出

Python の標準ライブラリである `re` は、正規表現（Regular Expression）を使って文字列の検索、抽出、操作を効率的に行うためのツールだ。正規表現を使うことで、調理動作のカウントを効率的に行うことができる。

- `re.search(pattern, string)`: 文字列中で最初に一致する部分を探し、そのマッチオブジェクトを返す。例えば、調理動作を示す「炒める」や「煮る」などの単語を探し、それが文章の中に含まれているかを調べる。
- `re.findall(pattern, string)`: 一致するすべての部分をリスト形式で返す。これを使うことで、献立の中に含まれるすべての調理動作（「炒める」「煮る」など）を抽出し、その頻度を数えることができる。
- `re.sub(pattern, repl, string)`: 一致する部分を指定した文字列に置き換える。この関数を使えば、例えば調理動作に関連する文字列を統一された形式に変換することができる（例えば、「焼く」→「グリル」）。

- `re.split(pattern, string)`: パターンに基づいて文字列を分割する。調理工程を細分化して、それぞれの調理動作を個別に抽出する際に有効だ。
- `re.match(pattern, string)`: 文字列の先頭部分がパターンに一致するかを確認する。これを利用することで、調理動作が文章の最初に登場する場合に処理できる。

正規表現の基本構成要素

正規表現は、特定の文字列パターンを定義するための手段であり、以下の基本的な構成要素で成り立っている：

- **リテラル文字**: 最も基本的な正規表現の構成要素は、文字そのものだ。例えば、"炒める"や"煮る"といった文字列をそのまま検索する。
- **メタ文字**: 特定の意味を持つ文字があり、これを使うことで検索パターンをより柔軟にする。例えば、`.` は任意の一文字、`*` は直前の文字が0回以上繰り返されることを意味する。
- **括弧（丸括弧）**: `()` を使って、グループ化したり、部分的に検索することができる。例えば、`(炒める|煮る)` は、「炒める」または「煮る」のいずれかに一致する。
- **角括弧（文字クラス）**: `[]` 内に複数の文字を列挙して、その中のいずれかの文字に一致するパターンを作成できる。例えば、`[あいお]` は「あ」または「い」または「お」に一致する。
- **繰り返し**: `*`（直前の文字が0回以上繰り返す）、`+`（直前の文字が1回以上繰り返す）、`?`（直前の文字が0回または1回繰り返す）などを使うことで、パターンを繰り返し検索できる。

正規表現のメリット

正規表現を使うメリットとして、以下の点が挙げられる。正規表現を使うことで、単純なキーワード検索に留まらず、「特定のパターンを持つ文字列」や「部分一致」にも対応できる。例えば、「炒める」や「煮る」を含む動作を一度に抽出することが可能だ。また、複数の調理動作を含む文章から、特定の動作を迅速に抽出することができ、大量のデータ処理にも適している。レシピや調理方法のテキストを処理することで、非構造的な「作り方」情報を構造化されたデータ（調理動作のカウント）に変換できる。これにより、モデルが処理できる形にデータを変換することができる。

このように、正規表現を活用することで、調理動作のパターンを自動的に抽出し、分析の準備を効率化することが可能になる。特に献立の情報から調理工程を解析する際に非常に有効だ。

栄養素量の単位削除

クレイピングした献立の情報には、栄養素量の単位（例えば、「g」や「ml」）が含まれていることがある。この単位をそのままにしておくと、数値として比較する際に問題が生じる。そのため、栄養素量の数値だけを抽出し、単位を削除した。これにより、異なる栄養素の量を単位に依存せずに比較でき、さらに数値データとして解析できるようになる。例えば、「100gの鶏肉」「200mlのスープ」といったデータから、単位を取り除き「100」「200」という純粋な数値のみを使用して分析を行う。

Python のデータ処理ライブラリ pandas には、DataFrame というデータ構造が用意されている。DataFrame は、表形式（行と列）でデータを管理できるオブジェクトであり、データの操作や解析を簡単に行うことができる。例えば、Excel のスプレッドシートに似た形式でデータを扱うことが可能で、行列ごとにアクセスしたり、データを加工したりすることができる。

特徴は行と列を持つ二次元構造で、数値や文字列などの異なる型のデータを含むことができる。大量のデータを効率的に操作し、フィルタリングや集計などが簡単に行えるので利便性が高い。

applymap メソッドを使ったデータ処理

DataFrame 内のデータを処理する際、applymap メソッドを使用することで、各セルに特定の関数を一括適用することができる。applymap メソッドは、DataFrame 全体に適用されるため、栄養素量の単位削除といった処理に適している。

使い方は DataFrame.applymap(func) の形で使用する。ここで func は、各セルに適用する関数を指定する。各セルに個別の処理を自動で適用できるため、大量のデータを効率的に処理する際に有用である。

applymap メソッドを利用することで、栄養素量から単位を削除する関数を簡単に適用し、数値データとして扱いやすく変換できる。

時間帯フラグの設定

最後に、献立が提供される時間帯（朝食、昼食、夕食など）を示すフラグを設定した。献立の提供時間に基づいて、それがどの時間帯に該当するかを判定し、時間帯ごとのフラグを立てた。例えば、朝食の献立には「朝食」フラグに 1 を、昼食の献立にはフラグに 2 を設定する。

このようにすることで、献立が時間帯に応じた特徴を持っているかどうかをモデルに学習させることができる。時間帯によって食事内容や栄養素、調理時間が異なる可能性があるため、時間帯を考慮した分析ができるようになる。

§ 3.3 ロジスティック回帰分析

ロジスティック回帰分析（Logistic Regression）は、従属変数（目的変数）が 2 つのカテゴリに分類される場合（バイナリ分類）に使用される統計手法である [?]。この手法は、入力変数（説明変数）と目的変数の間に線形関係がない場合でも、分類問題を解決するために広く利用される。

表 3.2: 処理したデータ一覧

グループ名	特徴量	特徴量数	変数タイプ
基本情報	'レシピの名前', '主菜フラグ', '調理時間', '摂取カロリー', 'コスト', '朝食'	6	量的: '調理時間', '摂取カロリー', 'コスト'; 質的: 'レシピの名前', '主菜フラグ', '朝食'
UX項目	'q1', 'q2', 'q3', 'q4'	4	質的
調理手法	'切_count', '混_count', '焼_count', '煮_count', '炒_count', '蒸_count', '揚_count', '茹_count', '和_count'	9	量的
栄養成分	'たんぱく質', '炭水化物', '糖質', '脂質', '食塩相当量', '食物繊維', 'ビタミンA', 'ビタミンB1', ...	30	量的
食材分類	'牛肉', '豚肉', '鶏肉', 'ハム', 'ソーセージ', 'ベーコン', '鮭', 'サーモン', 'ぶり', ...	40	質的

ロジスティック回帰の基本的なアイデアは、従属変数があるカテゴリに分類される確率を予測することである。回帰分析と名前はついていますが、実際には回帰モデルではなく、分類問題を扱うために使用される。従属変数は、0 または 1 の 2 つの値を取る確率変数であり、モデルの出力はその確率に関連付けられている。

ロジスティック関数

ロジスティック回帰では、ロジスティック関数（シグモイド関数）を用いて、モデルの出力を確率として解釈する。ロジスティック関数は、以下の式で表される。

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \dots + \beta_n X_n)}}$$

ここで、 $P(Y = 1|X)$ は、 X に基づいて $Y = 1$ となる確率、 e はネイピア数、 β_0 は切片、 $\beta_1, \beta_2, \dots, \beta_n$ は回帰係数、 $X_1, X_2, \dots, X_n$ は説明変数である。この関数は、入力された値に対して、出力が 0 から 1 の範囲に収束するため、確率として解釈可能となる。

回帰係数とオッズ比

ロジスティック回帰では、各説明変数の影響を回帰係数 β_j によって表現する。 β_j は、説明変数 X_j が 1 単位変化したときに、従属変数 $Y = 1$ となる確率のオッズ比がどれだけ変化するかを示す。

オッズ比は、あるイベントが発生するオッズ（確率/1-確率）の比率であり、ロジスティック回帰モデルにおいて重要な役割を果たす。オッズ比は、回帰係数 β_j を指数関数で変換することによって得られる。すなわち、

$$\text{オッズ比} = e^{\beta_j}$$

例えば、ある回帰係数 β_j が 1 である場合、そのオッズ比は $e^1 \approx 2.718$ となり、 X_j が 1 単位増加することで $Y = 1$ のオッズが約 2.7 倍になることを示す。逆に、 β_j が負であれば、オッズ比は 1 未満となり、 X_j の増加が $Y = 1$ のオッズを減少させることを意味する。

オッズ比を用いることで、各説明変数が目的変数に与える影響を直感的に理解することができる。オッズ比が 1 より大きければ、その変数が目的変数に対して正の影響を持ち、1 より小さければ負の影響を持つことを示す。

回帰係数の推定

ロジスティック回帰では、モデルのパラメータ（回帰係数）を最尤推定（Maximum Likelihood Estimation: MLE）によって推定する．最尤推定は、観測データが最も高い確率で得られるようなパラメータの値を求める方法であり、ロジスティック回帰においても広く用いられる．

最尤推定法では、以下の尤度関数を最大化することにより、 β を推定する．

$$L(\beta) = \prod_{i=1}^n P(y_i|x_i, \beta)^{y_i} (1 - P(y_i|x_i, \beta))^{1-y_i}$$

ここで、 y_i は観測された結果、 x_i は対応する説明変数の値、 $P(y_i|x_i, \beta)$ は、 x_i と β に基づく予測確率である．

本実験での目的変数

本実験では、推定する目的変数を以下の4つのUX項目とする．

1. 「食材や調味料が入手のしやすいか？」
2. 「普段では思いつかない料理か？」
3. 「朝や昼、夜などの時間帯にふさわしい料理か？」
4. 「調理ができそうな料理か？」

それぞれの選定理由について説明する

1. 入手のしやすさについて

入手のしやすさは、献立の実現可能性に大きく影響を与える要素である．日常的な調理において、必要な食材や調味料が手に入らない場合、レシピを選択しても実際に調理することが困難になる．そのため、「食材や調味料が入手しやすいかどうか」を考慮することは、実際の利用シーンを想定した献立作成において重要である．また、一般的にスーパーやコンビニで容易に手に入る食材を使用したレシピであれば、多くのユーザーにとって受け入れられやすく、実行可能性が高いと考えられる．

2. 普段では思いつかない料理かどうかについて

普段では思いつかない料理は、ユーザーに新しい食事の発見や体験を提供するための重要な要素である．日常的に似たような献立が繰り返されると、食事に対するモチベーションが低下する可能性がある．そのため、普段の選択肢にはない料理を提案することで、献立作成システムがユーザーに新鮮さや驚きを提供できる．また、新しい料理を取り入れることで、栄養バランスの改善や食生活の多様性向上にも寄与する．

3. 時間帯にふさわしい料理かどうかについて

朝、昼、夜といった食事の時間帯は、それぞれのシーンに適した料理が求められる．たとえば、朝食には調理が簡単で消化の良い料理、昼食には活動エネルギーを補給できるバランスの取れた料理、夕食にはカロリーが控えめでリラックスできる料理が選ばれることが多い．時間帯にふさわしい料理を選定することで、献立提案がより実用的で具体的なものとなり、ユーザーにとって受け入れやすい提案が可能になる．

4. 調理ができそうな料理かどうかについて

調理ができそうかどうかは、ユーザーのスキルや手間を考慮した現実的な要素である．どれだけ魅力的な料理であっても、調理工程が難解であったり、調理時間が過度に長かったりすると、実際には選ばれにくい．そのため、ユーザーのスキルや調理環境を考慮し、「この

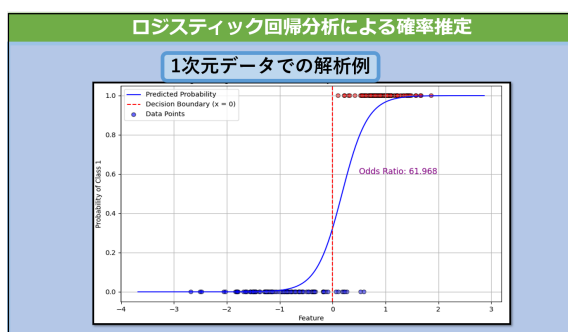


図 3.2: ロジスティック回帰分析のイメージ

		予測値	
		正	負
正解値	正	TP	FP
	負	FN	TN

図 3.3: 評価指標について

料理なら自分でも作れそうだ」と思える献立を提案することが重要である。これにより、提案した献立が実際の行動に結びつきやすくなり、献立作成システムの利用価値が向上する。

モデル評価

ロジスティック回帰分析のモデル評価には、主に以下の指標が使用される。

- **精度 (Accuracy)**: モデルが正しく予測したデータの割合。以下の式で求められる。

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (3.1)$$

ここで、TP は真陽性、TN は真陰性、FP は偽陽性、FN は偽陰性を表す。

- **適合率 (Precision)**: 正と予測されたデータの中で、実際に正であったデータの割合。以下の式で求められる。

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3.2)$$

- **再現率 (Recall)**: 実際に正であるデータの中で、正と予測された割合。以下の式で求められる。

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.3)$$

- **F 値 (F1-score)**: 精度と再現率の調和平均であり、以下の式で求められる。

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.4)$$

F1 値は精度と再現率のバランスを評価するために使用される。

また、ロジスティック回帰モデルの適合度を評価するためには、**擬似決定係数**や **AIC (赤池情報量基準)** などの指標が使用されることがある。これらの指標は、モデルの精度や過剰適合を防ぐために役立つ。

擬似決定係数は、回帰モデルがどの程度データを説明できているかを評価するための指標であり、通常は 0 から 1 の間の値を取る。この値が高いほど、モデルがデータをよく説明していることを示す。

一方で、AIC (赤池情報量基準) は、モデルの適合度と複雑さのバランスを評価する指標であり、値が小さいほど良いモデルとされる。AIC は、モデルが過剰適合していないかを判断するためにも重要な役割を果たす。

提案手法

§ 4.1 UX データの収集

本研究では、献立作成を多目的最適問題としてとらえる。目的関数を献立に含まれる料理の調理時間と調理にかかるコストの最小化とし、制約条件を必要栄養素や摂取カロリーなどとして多目的最適化を行う。そして、二つの目的関数の最小化と、複数の制約条件に基づいた、パレート最適である献立を出力する。なお、献立の日数や料理の準備にかかる時間などのユーザの選好がかかわる制約条件はユーザが選択できる。多目的最適化問題を解く手段として、NSGA-II という遺伝的アルゴリズムを多目的最適化に応用した手法を用いる。なお、このシステムは pymoo という Python プログラム用いて、NSGA-II によって組み合わせ最適化問題を解かせるよにプログラムの記述を行う。NSGA-II によって出力したパレート解である献立は、摂取栄養量やカロリー、主菜と副菜の数、アレルギー制限などの制約条件をみたし、調理にかかるコストおよび調理にかかる時間が最小化された、パレート最適な集合として複数出力される。この出力された複数の献立のうちユーザに最もあった献立をユーザ自身に選択してもらう。そして、その後 UX 項目についてユーザに回答してもらう。UX データ収集に使われる Flask について説明する。

Flask

Flask は、Python で書かれた軽量な Web アプリケーションフレームワークで、シンプルで拡張性に優れた設計が特徴。Flask は「マイクロフレームワーク」と呼ばれ、最小限のコア機能を提供し、その上に必要なライブラリや機能を追加していく形で柔軟に構築が可能。この特性により、開発者はプロジェクトのニーズに応じて自由に機能を追加したり、不要なものを省いたりでき、スケーラブルで効率的なアプリケーションの開発ができる。

Flask の基本的な特徴には以下の点がある

簡潔な設計: Flask は、シンプルな API と構造で、初心者でも理解しやすい。標準のテンプレートエンジンである Jinja2 や、URL ルーティングを容易に設定できる仕組みが組み込まれており、迅速な開発が可能。

拡張性: Flask は、必要に応じてデータベース、認証、フォーム処理、ファイルアップロード、セッション管理など、追加の機能を簡単に組み込むことができる。これにより、最初は小規模なアプリケーションとして開始しても、後に大規模なシステムへと成長させることができる。

テンプレートエンジン: Flask は Jinja2 という強力なテンプレートエンジンを使用して、HTML や XML などのドキュメントを動的に生成できる。これにより、ユーザーに対して動的に生成されるコンテンツを提供する Web アプリケーションの開発が可能。

デバッグ機能と開発サーバ: Flask は開発中に役立つデバッグモードを備えており、コードの変更を即座に反映できる開発サーバも内蔵。これにより、開発者は素早くフィードバックを得ることができ、アプリケーションの開発が効率的に行える。

軽量性と柔軟性: 他の Web フレームワークと比較して、Flask は非常に軽量で、無駄な機能を持たないため、最小限のリソースで動作する。そのため、非常に小さな Web アプリケーションから、大規模な API システムまで幅広い用途に対応できる。

Flask はそのシンプルさと拡張性から、プロトタイピングや小規模から中規模の Web アプリケーションの開発に非常に適しており、API を公開したり、データ分析の結果を Web 上で表示したりするシステムにもよく使用される。また、Flask の豊富なプラグインを活用することで、特定の機能を簡単に実装でき、開発スピードを大幅に向上させることができる。

このように、Flask は Web アプリケーション開発における強力なツールで、そのシンプルで柔軟な設計が多くの開発者に支持されている。

変数

各変数は、対象の日数を D 、日の番号を k 、レシピの数を R 、料理レシピが献立に含まれている場合に 1、含まれていない場合に 0 の値をとる献立フラグを r_{ki} 、料理レシピが主菜の場合に 1、副菜の場合に 0 の値をとる主菜フラグを σ_i 、 i 番目の料理レシピの調理時間を T_i 、 i 番目の料理レシピの食材コストを G_i 、 i 番目のレシピの UX 項目の推定確率それぞれを $Q1_i$ 、 $Q2_i$ 、 $Q3_i$ 、 $Q4_i$ 、 i 番目の料理レシピの l 番目の摂取栄養素を f_{il} 、 l 番目の栄養素の制約の最大値を F_l^H 、最小値を F_l^L 、 i 番目の料理レシピの摂取カロリーを C_i 、基礎代謝量の制約の最大値を B^H 、最小値を B^L 、朝食、昼食、夕食における最大調理時間をそれぞれ τ_1, τ_2, τ_3 とする。

また、入力画面でアレルギーが選択されていた場合に 1、選択されていない場合に 0 の値をとるアレルギーフラグを x_i 、各制限食が選択されていた場合に 1、選択されていない場合に 0 をとる制限食フラグを y_i 、制限食における栄養素の制約の最大値を E_l^H 、最小値を E_l^L とする。

本研究で提案する、自動献立作成システムにおける多目的最適化問題の目的関数と制約条件は、上記の変数を用いて下の式によって定式化される。

< 定式化 >

$$\text{minimize} \quad \sum_{k=1}^{3D} \sum_{i=1}^R r_{ki} T_i \quad (4.1)$$

$$\text{minimize} \quad \sum_{k=1}^{3D} \sum_{i=1}^R r_{ki} G_i \quad (4.2)$$

$$\text{minimize} \quad \sum_{k=1}^{3D} \sum_{i=1}^R r_{ki} Q1_i \quad (4.3)$$

$$\text{minimize} \quad \sum_{k=1}^{3D} \sum_{i=1}^R r_{ki} Q2_i \quad (4.4)$$

$$\text{minimize} \quad \sum_{k=1}^{3D} \sum_{i=1}^R r_{ki} Q3_i \quad (4.5)$$

$$\text{minimize} \quad \sum_{k=1}^{3D} \sum_{i=1}^R r_{ki} Q_{4i} \quad (4.6)$$

$$\text{subject to} \quad F_l^L \leq \sum_i^R r_{ki} f_{il} \leq F_l^H \quad (\forall k, \forall l) \quad (4.7)$$

$$B^L \leq \sum_i^R r_{ki} C_i \leq B^H \quad (\forall k) \quad (4.8)$$

$$\sum_i^R r_{ki} T_i \leq \tau_1 \quad (k \% 3 = 1) \quad (4.9)$$

$$\sum_i^R r_{ki} T_i \leq \tau_2 \quad (k \% 3 = 2) \quad (4.10)$$

$$\sum_i^R r_{ki} T_i \leq \tau_3 \quad (k \% 3 = 3) \quad (4.11)$$

$$0 < \sum_i^R r_{ki} \sigma_i \leq 1 \quad (\forall k) \quad (4.12)$$

$$0 \leq \sum_i^R r_{ki} (1 - \sigma_i) \leq 3 \quad (\forall k) \quad (4.13)$$

$$\sum_{k=1}^{3D} r_{ki} \leq 1 \quad (4.14)$$

$$0 \leq \sum_i^R r_{ki} x_i < 1 \quad (\forall k) \quad (4.15)$$

$$E_l^L \leq \sum_i^R y_i r_{ki} f_{il} \leq E_l^H \quad (\forall k, \forall l) \quad (4.16)$$

目的関数

本研究の献立作成における多目的最適化問題を構成する目的関数と制約条件式について説明する．まず，目的関数は，式 (4.1) とから式 (4.6) の 4 つであり，(4.1) は調理時間の最小化であり，(4.2) は食材コストの最小化，(4.3) から (4.6) は UX 項目の推定確率の最小化である．0-1 変数である献立フラグを用いて，設定した日数での料理の組み合わせを表現する．

制約条件

制約条件は，式 (4.7) から式 (4.16) の 10 つである．式 (4.7) は摂取栄養量制約，式 (4.8) は摂取カロリー量制約，式 (4.9) から式 (4.11) は朝食，昼食，夕食における最大の調理時間制約，式 (4.12) と式 (4.13) は主菜は 1 つ，副菜は 3 つ以下で献立を構成する制約，式 (4.14) は献立の中に，同じ料理が存在しないようにする制約である．また，式 (4.15) は，入力画面でアレルギーを選択した時に，そのアレルギーが含まれるレシピが含まれないようにする制約であり，式 (4.16) は，入力画面で制限食が選択されたときの摂取栄養素量の制約である．

摂取栄養量制約は，1 日あたりに摂取する特定の栄養量に，下限と上限を設定して表現する．摂取カロリー量制約は，1 日あたりに摂取するカロリー量に，下限と上限を設定して表

現する。朝食、昼食、夕食における最大の調理時間制約は、入力画面で入力した朝、昼、夜の各時間帯における献立にかかる調理時間をそれぞれ上限に設定した。

主菜は1つ、副菜は3つ以下で献立を構成する制約は、その料理が主菜であるか、副菜であるかを表現する 0-1 変数の主菜フラグを用いて、献立に含まれる主菜と副菜の下限と上限を表現する。

§ 4.2 UX を考慮した献立作成

UX を考慮した献立作成のために本システムでは遺伝的アルゴリズムを使用する。遺伝的アルゴリズム (Genetic Algorithm: GA) とは、近似解を探索するためのメタヒューリスティックアルゴリズムである [18]。メタヒューリスティクスアルゴリズムとは、特定の問題だけに限らず、どんな問題に対しても汎用的に対応できるように設計された、アルゴリズムの基本的な枠組みのことである。

GA は、解の候補であるデータサンプルを遺伝子で表した個体を複数体準備し、適応度関数によって計算された適応度の高い個体を優先して選択し、交叉、突然変異や淘汰などの操作を繰り返しおこなうことで最適解を導出する。GA は生命の進化過程に似ている様子からその名が付けられた。

次に、本研究で扱う手法である、非優越ソート遺伝的アルゴリズム (Non-dominated Sorting Genetic Algorithm: NSGA-II) について説明する。NSGA-II とは、Deb らによって 2002 年に提案された [19]、GA を、多目的最適化問題に拡張したものである。

NSGA は個体の評価方法を、Goldberg により提案された非優越ランキングソート [?] と、シェアリングを組み合わせたものを用いており、パレート最適的なアプローチによる手法の 1 つである。

非優越ソート (Non-Dominated Sort) とは、1989 年に Goldberg により提案されたアルゴリズムで、NSGA-II において適応度の高い個体を抽出するために用いられている個体のランク付けである。個体をランク付けし、同じランクの中でシェアリングを行う際に、ランクレベルのみでシェアリングを行うことによって、全個体についてシェアリングを行うよりも、計算にかかる負荷を軽減させることができる。

次に、多目的最適化で解いたパレート解のうち、ユーザにどのようにして最適な献立を出力させるかについて説明する。複数の目的関数の最小化または最大化を考える多目的最適化において、複数の目的関数を同時に満たすような解は存在せず、一方の目的関数が高い評価を得た場合、他方の目的関数は犠牲になってしまうトレードオフの関係になってしまうことが普通なため、目的関数が複数にある場合における解は、意思決定者にとって、最も好ましいものを選択できるようにすることが大事である。

対話型処理を用いたパレート最適解を選好する従来事例として、多目的最適化問題に関して、制約式と目的関数に含まれるパラメータの決定などの問題の設定時に含まれるあいまい性と、意思決定者があいまいな目標を持つことを考慮した、対話型ファジィ満足化手法がある。この手法では、個体の作成から最適化処理の部分はアルゴリズムが担い、その最適化処理の過程における評価の部分行っている。このシステムでは、ランダムで生成された個体をユーザに画像で提示し、提示された画像に対してユーザが 5 段階評価をし、その評価に従って近似最適解を再度作成している [20]。

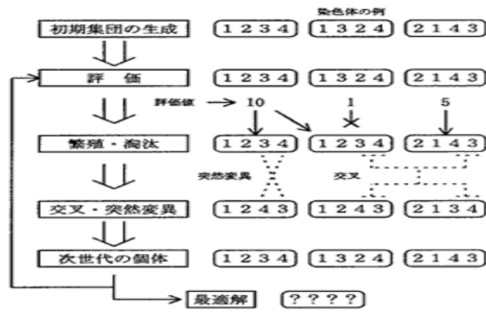


図 4.1: 遺伝的アルゴリズムのイメージ [23]

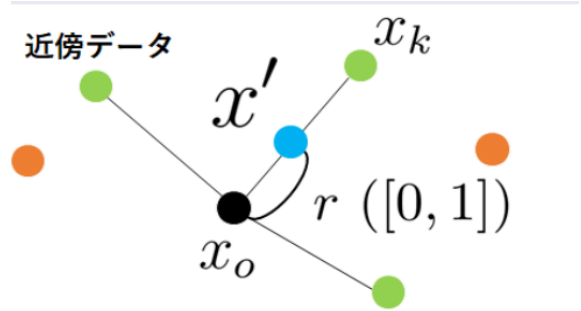


図 4.2: SMOTE のイメージ

また、単一目的の大規模な多目的離散最適化問題を、効率的に解を探索するためのアルゴリズムである、モジュラアプローチを用いて解き、それによって求められたパレート最適解集合の大きさを表示したのち、パレート最適解集合の大きさが決められた値以内になるまで繰り返しモジュラアプローチを用いて解き、縮小されたパレート最適解集合の中から各目的関数の重要度などの自分の選好条件に基づいて選好最適解を決定する手法が挙げられる。

他には、対話型 GA による近似最適解の探索を基本としつつも、GA の最適化処理の過程における、個体の適応度を評価をする部分を人間が行うといった手法も提案されている。一般的な GA での評価の役割は、評価関数が担っているが、対話型 GA では、この評価関数により個体の適応度を決定する部分を、人間が評価を行うようにしている。人間の意思決定を個体の適応度評価の過程に組み込むことにより、人間による主観的な評価が 1 つのシステムの要素となることから、対話型 GA は人の感性をシステムに落とし込むことが可能な手法である。

対話型 GA は、感覚や個人の好みなどといった、数値では表すことが困難な個人の感性を、対話型 GA による設計やデザインに取り入れることが可能となっているため、服飾やオフィスデザインや感性による様々な事柄への推薦、補聴器を使用する人の、聞こえに合わせるフィッティングなどへの研究に応用することが可能となっている。

意思決定者の選好解を求めるために、大きく分けて、以下の 3 つのアプローチがある。

1. 全て、もしくは十分に多くパレート最適解を求め、それを意思決定者に提示し、選好解を自分自身で決定してもらう。
2. 意思決定者の選好を表す実数値関数である、価値関数または効用関数を求め、それを最適化するような数理計画問題を解く。
3. コンピュータによって導出されたパレート最適解と、その解に基づく意思決定者の局所的な選好情報を用いて、ユーザとコンピュータの対話を繰り返すことによって、選好解を決定する。

最初のアプローチでは、目的関数の数が少ない場合や、実行可能解が少数で、有限個しか存在しない場合に有効であるとされる。この方法で代表的なものとして、各目的関数に対する重みを用いて、問題を解く加重和最小化や、1 つの目的関数を残し、他の目的関数に対する要求水準を制約条件に用いる、制約変換法などがある。

2 番目のアプローチの、価値関数もしくは効用関数の同定について、多属性効用理論が

知られており、目的関数間の独立性が十分確保されていることが重要となる。1 番目のアプローチで挙げた、加重和目的関数を、価値関数もしくは効用関数として想定して、そのパラメータを同定するといった、このアプローチの簡略版も考えられる。

最後のアプローチは、対話型解法と呼ばれており、意思決定者が、システムとの対話することによって、複数ある目的関数をどのように選り好みするかといった、局所的な選好情報を用いて、パレート最適解から解を自動的に選択する、という方法である。

この方法は、コンピュータとユーザの両者の情報交換の仕方によるので、いくつかの方法が考えられるが、ユーザという人間が関わっているということから、ヒューマンフレンドリーである方法が望まれる。提案されてきた対話型解法として、意思決定者が目的関数に対する、望ましいと考える値である希求水準を設定して、それに最も近い解をパレート最適解から得るという、希求水準法などが挙げられる。

本研究では、対話型処理によって、ユーザに対して分かりやすく献立を、選択してもらいたいと考えたため、3 番目のアプローチをとる。また、ユーザからのフィードバックをもとに、ロジスティック回帰分析を行う。しかし、回答データに偏りが生じた場合、不均衡なデータを扱うことになるため、分析結果の信頼性が低下する可能性がある。そこで、本研究では解析の前段階として SMOTE (Synthetic Minority Over-sampling Technique) を使用し、データのバランスを調整することで、モデルの予測精度を向上させることを目指す。SMOTE は少数派クラスのデータを合成的に増加させ、より均衡の取れたデータセットを作成する手法であり、不均衡データによる問題を軽減できると期待される。

SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) は、不均衡なデータセットに対処するための手法で、少数派クラスのサンプルを合成的に生成する方法である。具体的には、少数派クラスのデータポイント間で新たな合成サンプルを作成することで、データセットのバランスを取る。SMOTE は、各少数派クラスのデータポイントを選び、その近隣のサンプルとの線形補間を行って新しいデータポイントを生成する。この過程により、少数派クラスのデータ数を増やし、学習アルゴリズムが少数派クラスの特徴をより学習しやすくなる。

SMOTE の具体的な処理の流れは以下の通りである：

1. 少数派クラスの各サンプル $\mathbf{x}_i$ に対して、その近隣にある k 個のサンプル $\mathbf{x}_j$ を見つける。
2. そのサンプルとの間で線形補間を行い、新しいサンプルを生成する。新しいサンプル $\mathbf{x}_i^*$ は次の式で表される：

$$\mathbf{x}_i^* = \mathbf{x}_i + \lambda \cdot (\mathbf{x}_j - \mathbf{x}_i) \quad (4.17)$$

ここで、 λ は $[0, 1]$ の範囲でランダムに選ばれるスカラー値であり、補間の度合いを決定する。

3. 生成されたサンプル $\mathbf{x}_i^*$ を元のデータセットに追加し、バランスの取れたデータセットを形成する。

この手法の利点は、単に少数派クラスのデータを複製するのではなく、合成サンプルを生成する点にある。これにより、過学習を防ぎつつ、少数派クラスの特徴を豊富に学習させ

ることができる。しかし、SMOTE には注意が必要で、特に高次元データやノイズの多いデータでは、生成されたサンプルが効果的でない場合もある。そのため、適切なパラメータ設定や前処理を行うことが重要である。

本研究では、SMOTE を適用することで、不均衡なデータセットの問題を解消し、ロジスティック回帰モデルの精度を向上させることを目指す。これにより、ユーザからのフィードバックを基にした分析結果がより信頼性の高いものになると期待される。

§ 4.3 提案システムの流れ

本研究で提案する制限食と多人数考慮した自動献立作成システムの流れを図 4.3 に示す。また、本システムの流れを説明する。

Step 1: 料理レシピ、食材価格のデータベースの作成

まず、Python を用いて Web スクレイピングを行い、料理レシピと食材価格のデータベースを作成する。スクレイピングには、HTML 要素を検索してデータを抽出するために BeautifulSoup4 ライブラリを使用し、対象の Web ページから HTML を取得する際には Requests を使用する。レシピサイトとして、「ボブとアンジー」、「EatSmart」、「おいしい健康」の 3 つのサイトを活用する。これにより、1 つのレシピサイトに偏ることなく、幅広いレシピ情報を収集できる。また、食材価格情報は「小売物価統計調査による価格推移」を参照し、食材ごとの価格データを収集する。

収集したデータは、各料理レシピに含まれる栄養素、カロリー、調理時間、材料名、材料量、作り方、アレルギー情報などが含まれ、これらの情報は CSV ファイルとして保存される。これにより、同じ形式でレシピデータを扱うことができるようになる。食材価格データも同様に CSV ファイルに保存され、その後の計算に使用される。

食材名の一致を処理するために、Python のライブラリである difflib を利用して、レシピに記載された食材名と価格データベースの食材名の類似度を計算し、近い食材名を検索する。この手法でも一致しない場合は、楽天市場などのショッピングサイトを活用し、オンラインで食材の価格を取得して CSV に追加する。

Step 2: ユーザ情報と制約条件の入力

献立作成を行うためには、ユーザ情報を収集し、制約条件を設定する必要がある。ユーザ情報には、参加人数、身長、体重、年齢、性別、アレルギー、生活習慣病の有無などが含まれる。この情報を基に、各ユーザの栄養必要量や推奨摂取量を計算する。さらに、献立の出力日数や、朝・昼・夜の各時間帯にかける調理時間の上限を設定する。これにより、実際の制約に合わせて献立が生成される。

ユーザが入力した情報は、身体情報データベースに保存され、最適化処理時に使用される。例えば、年齢や性別から基礎代謝量 (BMR) を計算し、推定エネルギー必要量 (EER) を求める。この情報に基づき、最適化アルゴリズムはユーザに最も適した献立を提案できるようになる。

Step 3: NSGA-II による多目的最適化と最適な献立の出力

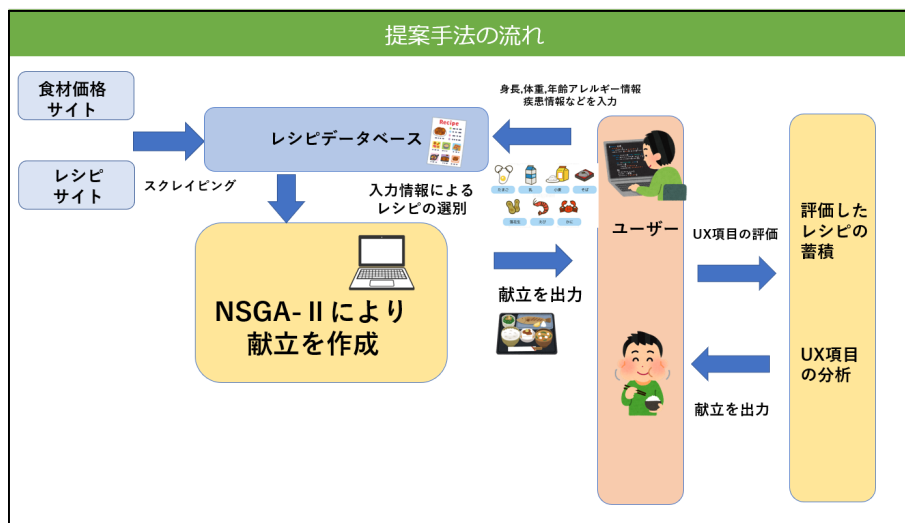


図 4.3: 提案手法の流れ

次に、料理レシピデータ群から入力された情報をもとに作成された制約条件と目的関数に沿った料理レシピを選択するという組み合わせ最適化問題と捉え献立作成を行う。献立作成に用いるデータまたは変数として、料理レシピサイトと食材とその価格を載せているサイトからスクレイピングしたレシピデータ、食材価格、栄養素データと、ユーザーによって入力された身長と体重、年齢、性別から計算された基礎代謝量、推定エネルギー必要量、アレルギー情報、疾患情報を用いる。これらの情報を多目的最適化問題を NSGA-II によって解き、パレート最適な献立を出力する。NSGA-II は、NSGA をエリート保存選択、混雑距離の導入、高速ソートの 3 点について変更と改良を施した手法であり、多目的最適化問題を解くアルゴリズムの 1 つである。目的関数には調理時間の最小化、使用する材料のコストの最小化が与えられ、制約条件には、3 大栄養素の摂取量、摂取カロリー量、朝、昼、夕の時間帯別の調理時間合計、献立に含まれる主菜と副菜の数、アレルギーがある場合にそのアレルギーの材料が含まれないようにする、疾患を患っている場合、その疾患にあった栄養素の摂取量の制限、出力される日数のうち、料理が被らないようにする、などの条件が設定されている。

Step 4: 対話型処理による献立の選択

ユーザーが重視するポイント（例：調理時間、コスト、UX 項目など）をスライダーを用いて調整し、その値に基づいて献立候補の優先順位を決定する。このシステムでは、スライダーの設定値が変化すると即座にアルゴリズムが再計算を行い、候補リストをリアルタイムで並べ替える。この動的な更新により、ユーザーは複数の優先条件を試行錯誤しながら献立をカスタマイズできる。

表示された候補リストには、各献立の詳細情報（例：調理時間、コスト、栄養素量、使用食材、UX 項目スコア）が含まれており、これにより選択肢を比較しやすい環境が提供される。また、候補のリストはウィンドウ上に視覚的に整理されて表示され、「決定」「もう一度決める」「終了」の 3 つの選択肢がユーザーに提示される。「もう一度決める」が選択された場合、再びスライダー画面に戻り、条件の調整を繰り返すことが可能である。これにより、ユーザーは試行錯誤を通じて、希望に最も合った献立を選択するプロセスを享受できる。

最終的に選ばれた献立に関しては、CSV ファイルから詳細な情報を取得し、各献立に関連する食材リスト、調理手順、栄養価情報などを詳細画面で表示する機能も備えている。このような対話型プロセスを通じて、ユーザーの意図を反映したカスタマイズされた献立候補を効率的に提供することが可能となる。

Step 5: 出力された献立のユーザーによる評価

システムによって出力された献立に対し、ユーザーは UX 項目の評価を行う。この際、評価項目は以下の 4 つで構成されており、それぞれについて Yes または No で回答を求める：

1. **入手しやすいか**（食材の購入が簡単かどうか） 2. **思いつかない料理か**（新しい提案性があるか） 3. **時間帯に沿った料理か**（朝昼晩に適した献立か） 4. **調理できそうか**（調理の実現可能性が高いか）

これらの評価結果は二値データとして CSV ファイルに記録される。この CSV には、献立に関連する詳細な情報（例：栄養素、調理時間、コスト、使用食材など）も統合されており、評価データと結びつけて次のロジスティック回帰分析に利用される。特にこの段階で得られたデータは、分析におけるテストデータとして活用されるため、ユーザーの評価データがシステムの改良に直結する構造を実現している。

Step 6: ロジスティック回帰分析

Step 5 で得られた評価結果を基にロジスティック回帰分析を行う。この分析では、表示されなかった献立における UX 項目の推定を行い、次回以降の献立推薦プロセスに活用する。この推定により、UX 項目（例：「調理できそうか」の Yes 回答率）が数値化され、目的変数として献立作成に反映可能となる。

分析に際しては、データの不均衡を補正するために SMOTE を用いる。特に「調理できそうか」や「思いつかない料理か」の評価において Yes/No の分布に偏りが生じる可能性が高いため、SMOTE の適用によりモデルのバイアスを軽減し、より正確な UX 項目の推定を実現する。また、この推定により、ユーザーが UX 項目を重視する際にスコアが高いと予測される献立を優先的に提示することが可能となる。

Step 7: UX 項目の目的関数への組み込み

ロジスティック回帰分析の結果を用いることで、従来の調理時間やコストといった目的変数に加え、UX 項目を目的関数として組み込むことが可能となる。このシステムでは、ユーザーが「調理時間」「コスト」「入手しやすいか」「思いつかない料理か」「時間帯に沿った料理か」「調理できそうか」の 6 つの選択肢から目的関数を 2 つ選び、多目的最適化を実行する。

これにより、UX 項目と既存の変数を統合的に考慮した献立作成が可能となり、ユーザーはより細かくパーソナライズされた推薦を受けられる。さらに、Step 3 から Step 7 のプロセスを繰り返すことで、システムはユーザーの評価データを活用し、個々のユーザーに適した献立推薦を提供し続ける。これにより、効率的かつユーザー満足度の高い献立作成プロセスを実現する。

数値実験並びに考察

§ 5.1 数値実験の概要

本研究の流れは料理レシピ、食材価格のデータベースの作成、ユーザ情報と制約条件の入力、NSGA-IIによる多目的最適化と最適な献立の出力、対話型処理による献立の選択、ユーザによるレシピの評価、ロジスティック回帰分析の結果を用いての献立作成の繰り返しとなっている。まず、使用するレシピデータ数は1000個とした。Pythonによるスクレイピングを行う際は、PythonのライブラリであるurllibとBeautifulsoup4を使った。

使用したレシピサイトは「ボブとアンジー」、「EatSmart」、「おいしい健康」の3種類からスクレイピングする。urllibにより、目的のレシピサイトと食材価格サイトのWebページのURLを渡し、そのページのHTML情報を取得したのちに、Beautifulsoup4を用いてWebページ上の料理レシピ名や摂取栄養素、食材とその価格などの必要な要素を、class名やid名などで指定し取得する関数を用いてスクレイピングを行う。Webサイトからのスクレイピングによって作成した料理レシピデータベースの例を図5.1に示す。

3つのレシピサイトからはスクレイピングする情報として、その料理から摂取することができる全栄養素やカロリー、調理時間、必要な材料名、材料量、料理のイメージ、アレルギー情報、作り方などをスクレイピングする。また各料理レシピの食材コストについては、料理に必要な食材と、食材価格データベースの中の食材を照らし合わせ、必要食材量と食材の価格、販売単位を用いて計算する。次に、NSGA-IIによる多目的最適化をしている際の実行画面を図5.2に示す。これはプログラムの内部で行われている処理を可視化したものであり、本研究はブラウザでシステムを用いている。そのため本研究では表示されない。NSGA-IIを用いた多目的最適化プログラムは、Pythonのライブラリである、pymooを利用して記述した。pymooは、多目的最適化や単目的最適化などの様々な解法をサポートを可能とするライブラリである。

今回の実験で設定した目的関数と制約条件について説明する。目的関数は、調理時間の最小化と、食材コストの最小化を設定する。制約条件は、健常者の場合と制限食が必要な人の場合で異なる。まず、健常者の場合について説明する。摂取栄養素については、3大栄養素である、たんぱく質、脂質、炭水化物のそれぞれに、1日に最低でも摂取すべき量を摂取できるように設定した。設定した値は、それぞれの3大栄養素に対して、たんぱく質は1日に必要な推定エネルギーの13%以上、脂質は15%以上、炭水化物は40%以上である。超えて摂取すると、健康障害のリスクが高まると定義される耐容上限量は、3大栄養素に関しては設定されていないため[21]、制約条件として上限値は設定しないことにした。

摂取カロリーについては、1日に必要なエネルギー量の目安を掲載している農林水産省の

エビと豆腐の煮物	1	10	174	むきエビ	120g	たんぱく質	15.9g
エビと豆腐の煮物	1	10	174	納豆し豆腐	1/2丁	炭水化物	6.4g
エビと豆腐の煮物	1	10	174	グリーンピー	大さじ1	脂質	5.8g
エビと豆腐の煮物	1	10	174	調味料等		脂質	8.5g
エビと豆腐の煮物	1	10	174	酒	大さじ1/2	食塩相当量	1.5g
エビと豆腐の煮物	1	10	174	だし汁	2/3カップ	食物繊維	0.6g
エビと豆腐の煮物	1	10	174	(A)		ビタミンA	4μg
エビと豆腐の煮物	1	10	174	・砂糖	小さじ1	ビタミンB0.12mg	
エビと豆腐の煮物	1	10	174	・うすくち	大さじ3/4	ビタミンB0.09mg	
エビと豆腐の煮物	1	10	174	水溶き片栗粉	適量	ビタミンB0.13mg	
エビと豆腐の煮物	1	10	174	サラダ油	適量	ビタミンB0.9μg	
エビと豆腐の煮物	1	10	174	NAN	NAN	ビタミンC2mg	
エビと豆腐の煮物	1	10	174	NAN	NAN	ビタミンD0μg	
エビと豆腐の煮物	1	10	174	NAN	NAN	ビタミンE1.9mg	
エビと豆腐の煮物	1	10	174	NAN	NAN	ビタミンK21μg	
エビと豆腐の煮物	1	10	174	NAN	NAN	カリウム351mg	
エビと豆腐の煮物	1	10	174	NAN	NAN	カルシウム82mg	
エビと豆腐の煮物	1	10	174	NAN	NAN	鉄1.4mg	
エビと豆腐の煮物	1	10	174	NAN	NAN	葉酸49μg	
エビと豆腐の煮物	1	10	174	NAN	NAN	マグネシウム67mg	
エビと豆腐の煮物	1	10	174	NAN	NAN	リン248mg	
エビと豆腐の煮物	1	10	174	NAN	NAN	亜鉛1.1mg	
エビと豆腐の煮物	1	10	174	NAN	NAN	マンガン0.38mg	
エビと豆腐の煮物	1	10	174	NAN	NAN	銅0.33mg	
エビと豆腐の煮物	1	10	174	NAN	NAN	マンガン0.33mg	

図 5.1: 栄養素データの例

n_gen	n_eval	cv (min)	cv (avg)	n_nds	eps	indicator
1	400	1.05290E+03	1.64265E+03	1	-	-
2	800	8.49100E+02	1.34299E+03	1	2.47000E+02	idea
3	1200	8.49100E+02	1.23941E+03	1	0.00000E+00	f
4	1600	8.20700E+02	1.16736E+03	1	3.16300E+03	idea
5	2000	7.87800E+02	1.09891E+03	1	2.02200E+03	idea
6	2400	7.79000E+02	1.03477E+03	1	4.85000E+02	idea
7	2800	7.16900E+02	9.78273E+02	1	8.84000E+02	idea
8	3200	5.68000E+02	9.29777E+02	1	1.58300E+03	idea
9	3600	4.22400E+02	8.85090E+02	1	1.97600E+03	idea
10	4000	4.22400E+02	8.42549E+02	1	0.00000E+00	f
11	4400	4.22400E+02	8.03732E+02	1	0.00000E+00	f
12	4800	4.22400E+02	7.62277E+02	1	0.00000E+00	f
13	5200	4.22400E+02	7.24457E+02	1	0.00000E+00	f
14	5600	4.22400E+02	6.92382E+02	1	0.00000E+00	f
15	6000	4.22400E+02	6.63149E+02	1	0.00000E+00	f
16	6400	3.96300E+02	6.33423E+02	1	1.19800E+03	idea
17	6800	3.96300E+02	6.06926E+02	1	0.00000E+00	f
18	7200	2.86900E+02	5.78641E+02	1	4.70000E+02	idea
19	7600	2.86900E+02	5.55043E+02	1	0.00000E+00	f
20	8000	2.86900E+02	5.31442E+02	1	0.00000E+00	f
21	8400	2.86900E+02	5.10590E+02	1	0.00000E+00	f
22	8800	2.86900E+02	4.93082E+02	1	0.00000E+00	f
23	9200	2.86900E+02	4.75208E+02	1	0.00000E+00	f
24	9600	2.86900E+02	4.57897E+02	1	0.00000E+00	f
25	10000	2.38800E+02	4.42169E+02	1	2.76000E+02	idea

図 5.2: 最適化処理の実行画面

サイト [22] を参考にして、基礎代謝量と身体活動レベルの係数をかけ合わせたものを使用した．そのため、上限値は 2536 キロカロリーに設定した．

次に、制限食が必要な人の制約条件について説明する．本研究で対象となる制限食が必要な人は、アレルギーを持っている人と、生活習慣病を患っている人である．また、対象となる生活習慣病は糖尿病、腎臓病、脂質異常症、高血圧とする．

まず、糖尿病を患っている人についてだが、4.2 章で述べた通り、糖尿病は、内臓脂肪型肥満によってインスリン抵抗性により発症する．そのため糖尿病の予防と改善には脂肪の是正が重要となってくる．また、厚生労働省によると、1 日あたりの炭水化物摂取量を 100 g 以下とする炭水化物制限が、肥満の是正に有効だとし、糖尿病の予防に有効だとしている．また、食物繊維の 1 日の平均摂取量が 20g を超えた時点から糖尿病の発症リスクに有意な低下傾向が見られている [10]．そのため、本研究における糖尿病の患者に対する制約条件として、エネルギー量、タンパク質摂取量は健常者と同じだが、1 日の炭水化物摂取量、脂質の摂取量、食物繊維の摂取量をそれぞれ 100g 以下、必要推定エネルギーの 15～25%，20g 以上とする．

次に、腎臓病を患っている人については、4.2 章で述べた通り腎臓病はたんぱく質制限、塩分制限、カリウム制限などの食事療法を行うことにより、腎機能障害の進行を抑え、慢性腎臓病の合併症を予防することができる．具体的な数値として日本腎臓学会によると 1 日のタンパク質の摂取量を標準体重当たり 0.6～0.7g とし、塩分の 1 日の摂取量は 3g 以上 6g 未満とし、カリウムの 1 日の摂取量が 1500mg 以下に制限することが推奨されている [11]．そのため、本研究における腎臓病の患者に対する制約条件として、エネルギー量、脂質、炭水化物の摂取量は健常者と同じだが、1 日のタンパク質と塩分と、カリウムの摂取量をそれぞれ標準体重当たり 0.6～0.7g、3g 以上 6g 未満、1500mg 以下とする．

脂質異常症を患っている人については、4.2 章で述べた通り脂質異常症は、コレステロール、食物繊維、脂質の摂取量を調整することにより脂質異常症の予防と改善に役に立つとされている [13]．本研究における具体的な制約条件としては、エネルギー量、炭水化物、タンパク質の摂取量は健常者と同じだが、1 日のコレステロール、脂質、食物繊維の摂取量をそれぞれ 200mg 以下、総エネルギーの 15%未満、20g 以上にする．

高血圧と診断されている人は、4.2 章で述べた通り高血圧の要因として、塩分を過剰に摂取することによる血圧上昇が大きな要因となるため、塩分制限が必要となっている．日本高血圧学会による「高血圧治療ガイドライン 2019」によると、高血圧者の減塩目標を食塩 6 g/日未満としている．また、カリウム摂取量増加によって高血圧者にとって血圧低下効

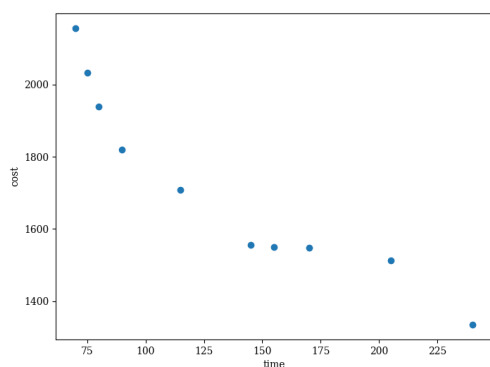


図 5.3: パレート解の出力

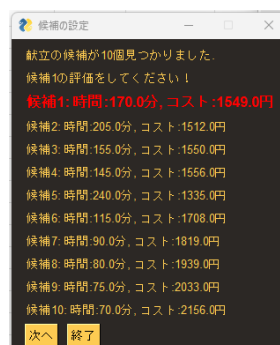


図 5.4: 対話型処理による解の選択

果を認めた。厚生労働省によると、カリウムの摂取量を 3510mg 以上摂取することが推奨されている。さらに、1 日の食物繊維の摂取量を 20g 以上にすること推奨されている [14]。以上のことから、本研究における高血圧者に対する制約条件として、エネルギー量、脂質、炭水化物、タンパク質の摂取量は健常者と同じだが、1 日のと塩分、カリウム、食物繊維の摂取量をそれぞれ 6g 未満、3510mg 以上、20g 以上とする。

次に、NSGA-II による最適化を行っている際の、実行画面について説明すると、 n_gen は現在の世代数、 n_level はこれまでの個体を評価した数、 $cv (min)$ 、 $cv (avg)$ はそれぞれ現在の母集団における最小の制約違反、現在の母集団における平均の制約違反、 n_nds は多目的最適化問題の場合の非劣解の数、 eps は過去数世代にわたるインジケータの変化、 $indicator$ はパフォーマンスインジケータを表す。

次に、NSGA-II による最適化処理が終わり、パレート最適解が出力された様子を図 5.3 に示す。縦軸は、指定した日数分の献立の合計調理時間を表しており、横軸は指定した日数の合計の食材コストを表している。

次に、パレート最適解から対話型処理を用いて献立を選択する画面を図 5.4 に示す。図 5.4 は、図 5.3 で表示されたパレート解を数値として表示する画面である。ユーザは、画面上の選択ボタンを使用して提示された候補を選択することができる。「献立を表示する」ボタンをクリックすると、選択した候補に対応する献立が出力される。その後、ユーザは各レシピについて 4 つの UX 項目を評価し、ロジスティック回帰分析を行うことで、UX 項目を考慮した献立作成が可能となる。

本研究では、以下の数値実験を実施する。まず、ユーザに身体情報を入力してもらい、時間とコストを目的関数とした最適化問題を解き、その結果得られた献立を提示する。次に、その献立に対して UX 項目に関する質問を行い、得られた評価データを CSV ファイルに保存する。このデータを基にロジスティック回帰分析を行い、ユーザごとに重要視する UX 項目を抽出する。さらに、UX 項目を考慮した献立を再度作成し、ユーザに評価を依頼する。このプロセスを繰り返すことで、ユーザが重要視する UX 項目を反映した献立が徐々に洗練されていく仕組みとなっている。また、ロジスティック回帰分析を行うごとにモデルの評価を行う。

表 5.1: モデル評価 (精度) の出力結果

精度	推薦前	推薦後	変化率
q1	0.42857	0.59524	0.38889
q2	0.49206	0.77778	0.58065
q3	0.36508	0.5873	0.6087
q4	0.66667	0.7619	0.14286
平均値	0.4881	0.68056	0.43027

表 5.2: 精度の t 検定の結果

	推薦前	推薦後
平均	0.4881	0.68056
分散	0.01686	0.01068
観測数	4	4
ピアソン相関	0.78149	
仮説平均との差異	0	
自由度	3	
t	-4.7501	
P(T<=t) 片側	0.00885	
t 境界値 片側	2.35336	
P(T<=t) 両側	0.0177	
t 境界値 両側	3.18245	

表 5.3: モデル評価 (適合率) の出力結果

適合率	推薦前	推薦後	変化率
q1	0.41667	0.75304	0.80729
q2	0.60684	0.88442	0.45742
q3	0.53704	0.75966	0.41455
q4	0.79206	0.85307	0.07702
平均値	0.58815	0.81255	0.43907

表 5.4: 適合率の t 検定の結果

	推薦前	推薦後
平均	0.58815	0.81255
分散	0.02465	0.00438
観測数	4	4
ピアソン相関	0.72211	
仮説平均との差異	0	
自由度	3	
t	-3.79	
P(T<=t) 片側	0.01611	
t 境界値 片側	2.35336	
P(T<=t) 両側	0.03222	
t 境界値 両側	3.18245	

§ 5.2 実験結果と考察

今回数値実験をするにあたって、朝、昼、夜の合計の調理時間の合計と主菜、副菜の数を共通にしておく。具体的な数値として、朝、昼、夜の合計の調理時間の合計をそれぞれ 15 分、45 分、60 分とし、主菜、副菜の数を 1 つ、2 つとした。また、以下は今回の数値実験に使用する 1 日に必要なエネルギーである、必要推定エネルギー量の計算式である。

被験者は大学生 21 才から 22 才の男性 6 名に対して実験を行った。それぞれにロジスティック回帰分析の後の目的関数をユーザ 1 は q1, q2, ユーザ 2 は q1,q3, ユーザ 3 は q1,q4, ユーザ 4 は q2,q3, ユーザ 5 は q2,q4, ユーザ 6 は q3,q4 を選択してもらった。アレルギーを持つ人はいなく、ロジスティック回帰分析を行うためのテストデータを集めるために目的関数に時間とコストが設定された献立を 1 週間分提示して評価を行ってもらった。その後ロジスティック回帰分析を行い、UX 項目の推定、考慮された献立の評価を行う。分析と評価を 3 回繰り返し、ロジスティック回帰分析前後の精度、適合率、再現率、F 値についての変化を考察する。

ユーザごとに重要視する UX 項目を選択してもらい、それに基づく評価を収集することで、モデルの精度向上を図る。本研究では、ユーザが繰り返し評価を行うプロセスを通じて、モデルが出力する献立が UX 項目においてより高い評価を得られるようになる過程を示す。さらに、一回目のモデル評価結果と四回目のモデル評価結果について対応のある t 検定を実施し、有意差が認められるかどうかを検証する。

本プログラムのモデルの考察

モデル評価項目ごとの結果とそれらの t 検定の結果を表 5.1 から 5.8 まで、ユーザごとの

表 5.5: モデル評価 (再現率) の出力結果

再現率	推薦前	推薦後	変化率
q1	0.83766	0.82108	-0.0198
q2	0.52894	0.85	0.60699
q3	0.39145	0.65694	0.67822
q4	0.74164	0.8634	0.16417
平均値	0.62492	0.79786	0.3574

表 5.6: 再現率の t 検定の結果

	推薦前	推薦後
平均	0.62492	0.79786
分散	0.04087	0.00914
観測数	4	4
ピアソン相関	0.69836	
仮説平均との差異	0	
自由度	3	
t	-2.2798	
P(T<=t) 片側	0.05348	
t 境界値 片側	2.35336	
P(T<=t) 両側	0.10695	
t 境界値 両側	3.18245	

表 5.7: モデル評価 (F 値) の出力結果

F 値	推薦前	推薦後	変化率
q1	0.65518	0.77718	0.1862
q2	0.52983	0.864	0.6307
q3	0.45208	0.69818	0.54439
q4	0.76441	0.85352	0.11657
平均値	0.60038	0.79822	0.36946

表 5.8: F 値の t 検定の結果

	推薦前	推薦後
平均	0.60038	0.79822
分散	0.01896	0.00595
観測数	4	4
ピアソン相関	0.56868	
仮説平均との差	0	
自由度	3	
t	-3.4933	
P(T<=t) 片側	0.01984	
t 境界値 片側	2.35336	
P(T<=t) 両側	0.03967	
t 境界値 両側	3.18245	

モデル評価の回数ごとの推移を図 5.5 から図 5.10 まで示す。本研究では、UX 項目を考慮した献立推薦の効果を検証するため、ロジスティック回帰分析の結果を用いて、推薦の前後で精度、適合率、再現率、および F 値を比較し、片側の t 検定を実施した。帰無仮説として「UX 項目を考慮した推薦によって指標に有意な差が生じない」を設定し、対立仮説として「UX 項目を考慮することで指標が改善される」を設定した。その結果、再現率を除く 3 つの指標（精度、適合率、F 値）で帰無仮説が棄却され、有意差が認められた。

まず、精度、適合率、および F 値が有意に向上した点は、UX 項目を考慮することで推薦アルゴリズムの性能が改善し、ユーザーの期待やニーズに応じた献立選択が可能になったことを示している。特に適合率の向上は、推薦結果の精度が高まり、誤った推薦が減少したことを示唆している。これにより、UX 項目の考慮がユーザー体験の向上に寄与する可能性が示唆される。

一方、再現率では有意差が認められなかったものの、p 値は 0.5348 と比較的惜しい結果となった。これは、UX 項目を考慮することで、ある程度再現率に影響を及ぼした可能性を示唆しているが、その効果が統計的に有意となるほどの強さではなかったと考えられる。再現率は、推薦システムがどれだけ多くの適切な項目を網羅的に捕捉できるかを測る指標であるため、UX 項目の導入がこの側面に与える影響が限定的であった可能性がある。

この結果を踏まえ、UX 項目が精度、適合率、および F 値に対して顕著な効果をもたらした一方で、再現率においてはさらなる改善の余地があると考えられる。

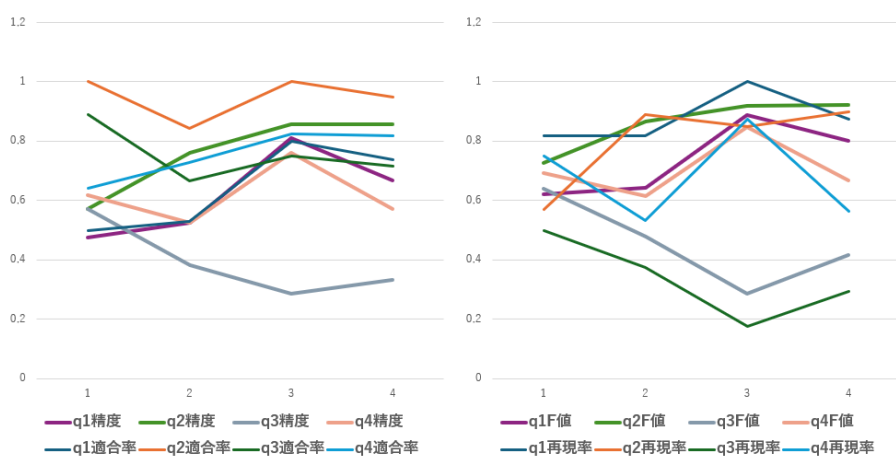


図 5.5: ユーザ 1 のモデル評価の推移

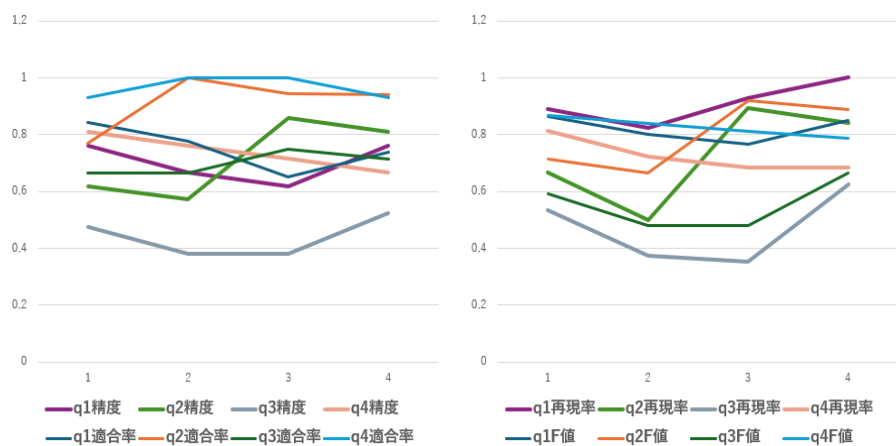


図 5.6: ユーザ 2 のモデル評価の推移

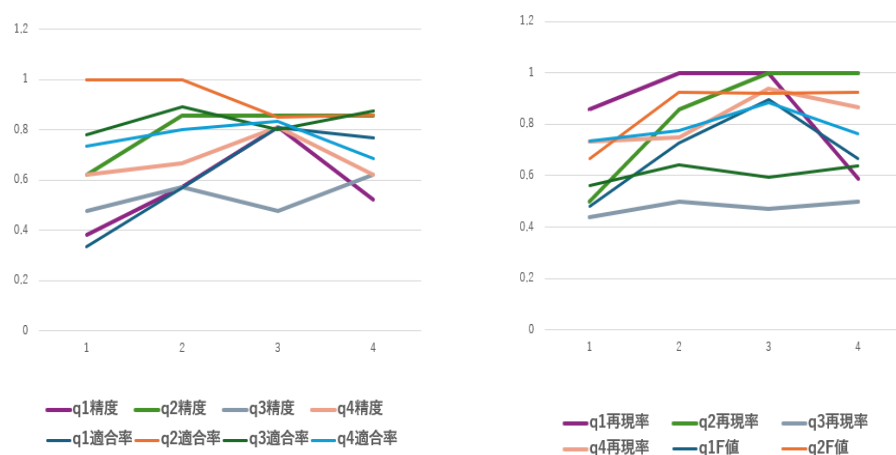


図 5.7: ユーザ 3 のモデル評価の推移

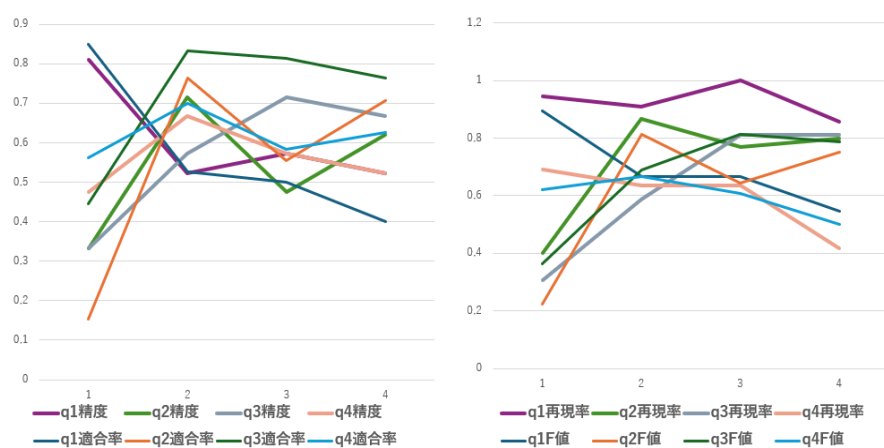


図 5.8: ユーザ 4 のモデル評価の推移

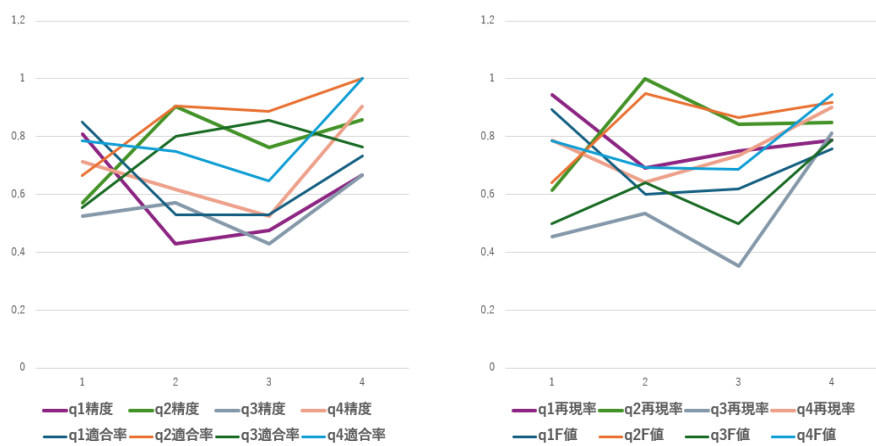


図 5.9: ユーザ 5 のモデル評価の推移

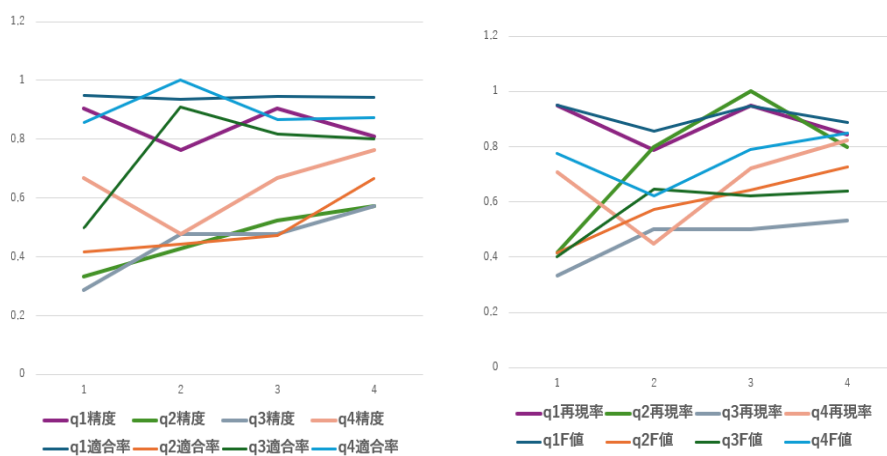


図 5.10: ユーザ 6 のモデル評価の推移

おわりに

本研究では、急激な生活様式の欧米化に伴い、ジャンクフードなどの過剰なエネルギー摂取を伴う食生活が広まり、生活習慣病の罹患率が増加しているという現状を踏まえ、生活習慣病の予防に向けたアプローチとして、栄養バランスの取れた食事の重要性に着目した。栄養バランスの取れた献立を作成することは、健康的な生活を送る上で非常に重要であるが、実際にはその作成には個々の身体情報や疾患情報を元にメニューや料理の分量を調整する必要がある、その作業は非常に手間がかかるものである。

このような問題を解決するために、本研究では、Webサイトから得られるレシピ情報や食材の価格などを活用し、ユーザーの健康状態や個々のニーズに合わせて栄養バランスを考慮した献立を自動的に作成できるシステムを提案した。具体的には、制約条件を考慮できる多目的遺伝的アルゴリズムを用いて、個々のユーザーに最適な献立を生成し、その過程でロジスティック回帰分析を活用して、UX項目を反映させることができるシステムを設計した。このシステムにより、ユーザーの健康状態や好みに応じて柔軟に献立を調整できるようになり、より個別化された食事提案が可能となる。

本研究で使用したレシピデータは、3つの異なるレシピサイトからスクレイピングを行い、データベースの多様性を持たせることに成功した。これにより、システムが提供する献立の選択肢が豊富になり、ユーザーの好みに応じた食事の提案が可能となった。また、このシステムは健常者だけでなく、生活習慣病やアレルギーを持つ人々にも対応できるように設計されており、幅広いユーザー層に対応した健康管理支援を提供できる点が大きな特徴である。

実際にシステムを動作させた実験結果に基づき、帰無仮説として「UX項目を考慮した推薦によって指標に有意な差が生じない」を設定し、対立仮説として「UX項目を考慮することで指標が改善される」を立てて検証を行った。その結果、再現率を除く3つの指標（精度、適合率、F値）について帰無仮説が棄却され、いずれの指標でも有意差が認められた。これにより、UX項目を考慮した献立推薦が、ユーザーにとってより精度の高い、適切な献立を提供することに寄与していることが示された。

この結果から、本研究の提案するシステムは、献立作成における最適化を実現し、単なる画一的な献立にとどまらず、ユーザーの身体情報やUX項目を考慮した個別化された提案を行えることが明らかになった。これにより、ユーザー一人ひとりの健康状態に適した献立の作成が可能となり、より効果的な健康管理が支援されることが期待される。

一方で、本研究にはいくつかの課題も残されている。例えば、摂取栄養素や摂取カロリーに関する上限・下限の設定など、制約条件をユーザーが柔軟に調整できるようにすることが今後の改良点として挙げられる。

謝辞

本研究を遂行するにあたり，多大なご指導と終始懇切丁寧なご鞭撻を賜った富山県立大学工学部情報工学科情報基盤工学講座の奥原浩之教授，António Oliveira Nzinga René 講師に深甚な謝意を表します．最後になりましたが，多大な協力をしていただいた研究室の同輩諸氏に感謝致します．

2025 年 2 月

堀由隆

参考文献

- [1] ソフトム株式会社, “ソフトム通信 第 79 号「給食業界における A I 活用」”, https://data.nifcloud.com/blog/food-service-provider_ai-use-case_01/, 閲覧日 2024.12.28.
- [2] C. A. Coello Coello and M. S. Lechuga, “MOPSO: a proposal for multiple objective particle swarm optimization, ” *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*, Vol. 2, pp. 1051-1056, 2002.
- [3] Q. Zhang and H. Li, “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition”, *IEEE Trans. Evolutionary Computation*, Vol. 11, No. 6, pp. 712–731, 2007.
- [4] LeftLetter, “多目的進化型アルゴリズム MOEA/D とその改良手法”, <https://qiita.com/LeftLetter/items/a10d5c7e133cc0a679fa>, 閲覧日 2025.1.6.
- [5] 株式会社おいしい健康, “おいしい健康”, <https://oishi-kenko.com/>, 閲覧日 2023.10.16.
- [6] 総務省統計局, “小売り物価統計調査による価格調査”, <https://jpmarket-conditions.com/>, 閲覧日 2024.10.11.
- [7] John H. Holland, “Adaptation in Natural and Artificial Systems”, 1975.
- [8] 厚生労働省, “日本人の食事摂取基準 (2020 年度版)”, <https://www.mhlw.go.jp/content/10904750/000586559.pdf>, 閲覧日 2025.01.15
- [9] 東京医科大学病院, “カリウムは調理のくふうで減らせます”, 内臓内科, <https://articles.oishi-kenko.com/syokujinokihon/dialysis/05/>, 閲覧日 2025.01.15
- [10] 厚生労働省, “糖尿病”, <https://www.mhlw.go.jp/content/10904750/000586592.pdf>, 閲覧日 2025.01.17
- [11] 厚生労働省, “慢性腎臓病”, <https://www.mhlw.go.jp/content/10904750/000586595.pdf>, 閲覧日 2025.01.17
- [12] 腎臓内科, “慢性腎臓病の食事療法”, 東京女子医科大学, <https://www.twmu.ac.jp/NEP/shokujiryouthou.html>, 閲覧日 2025.01.17
- [13] 厚生労働省, “脂質異常症”, <https://www.mhlw.go.jp/content/10904750/000586590.pdf>, 閲覧日 2025.01.17
- [14] 厚生労働省, “高血圧”, <https://www.mhlw.go.jp/content/10904750/000586583.pdf>, 閲覧日 2025.01.17
- [15] 厚生労働省, “食べ物アレルギー”, アレルギーポータル, <https://allergyportal.jp/knowledge/food/>, 閲覧日 2025.01.17

- [16] , “野菜一覧”,
<https://million-store.com/yasaiblog/?srsltid=AfmBOoq6J92TTZ8YEW1KXV5BqhavqWoxkSh8zC>,
閲覧日 2025.1.22.
- [17] エバラ食品, “食材一覧”,
<https://www.ebarafoods.com/recipe/clafoods/>, 閲覧日 2025.1.22.
- [18] John H. Holland, “Adaptation in Natural and Artificial Systems”, 1975.
- [19] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, “A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II”, *IEEE Tran. on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197, 2002.
- [20] 和正敏, “多目的線形計画問題に対する対話型ファジィ意思決定手法とその応用”, 電子情報通信学会論文誌 Vol. J 65-A, No. 11, pp. 1182-1189, 1982.
- [21] 厚生労働省, “日本人の食事摂取基準 (2020 年版) ”,
<https://www.mhlw.go.jp/content/10904750/000586553.pdf>, 閲覧日 2024.12.26.
- [22] 農林水産省, “一日に必要なエネルギー量と摂取の目安”,
https://www.maff.go.jp/j/syokuiku/zissen_navi/balance/required.html, 閲覧日 2025.1.22.
- [23] 電力土木技術協会, “遺伝的アルゴリズム”,
<https://www.jepoc.or.jp/tecinfo/library.php>, 閲覧日 2025.1.22.