できるビジネス

ビジネスの現場で使える

AI&データサイエンスの全知識

機械学習&ディープラーニング入門にも最適

データサイエンティスト
三好大悟 著

- データの集計・可視化・基礎統計
- 機械学習
- 画像認識
- レコメンデーション
- 数理最適化

これからの必須スキル

わかりやすいカラー図版

すぐ使えるExcelサンプル付き!

インプレス

できるビジネス

ビジネスの現場で使える

AI&データサイエンスの全知識

機械学習&ディープラーニング入門にも最適

データサイエンティスト
三好大悟 著

- データの集計・可視化・基礎統計
- 機械学習
- 画像認識
- レコメンデーション
- 数理最適化

これからの必須スキル

わかりやすいカラー図版

すぐ使えるExcelサンプル付き!

インプレス

はじめに

本書執筆中の 2021 年 9 月、デジタル庁が設置されました。デジタル庁は国・地方行政の IT 化や DX（デジタルトランスフォーメーション）の推進を目的とした省庁です。国をあげて、IT 化や DX への機運が高まっている中、民間企業においても、IT 化・DX の重要性は高まってきています。

そして、私たちの IT スキルの向上も重要になっていると同時に、その IT により蓄積されていくデジタルデータをいかに活用していくかという観点から、データの活用スキルも重要となってきます。しかし、データを活用するスキルを持った人材（データアナリスト、機械学習エンジニア、データサイエンティストなど）の需要が高まる反面、供給は不足しており、需給のギャップは今後さらに広がっていくと指摘されています。学習指導要領では、高校卒業までに情報活用能力を育成することが目標とされ、2023 年度より高校で始まる「情報 II」では、AI・データサイエンスの項目が盛り込まれました。国をあげて AI・データサイエンスリテラシーを高めていくという方針が示され、実行されつつあるのが昨今の状況です。

また、ビジネスに目を向けると、AI・データサイエンスの実装スキルまでは求められていないにせよ、そのようなスキルを持つ機械学習エンジニアやデータサイエンティストといった人材を適切にハンドリングし、プロジェクトを推進していく立場の人材も足りていないと感じています。

そのような背景もあり、自分自身や自社の業務範囲で AI・データサイエンスをどうビジネスに適用するかについて、ぜひ多くの方に知ってもらいたいというのが、本書を執筆する 1 つのきっかけでした。

本書の読者対象として、たとえば AI/DX 推進部署に配属された方々や、データ活用がミッションとして掲げられた部署の方々、データ活用や AI/DX の推進、またそのビジネス適用や社会実装に興味がある方々などはまさにぴったり当てはまるでしょう。もちろんそれ以外の方々にとっても役立つ場面は多々あるでしょう。 少しでも興味がある方はぜひ手にとっていただけると幸いです。

本書は、さまざまなビジネスケースをベースとしつつも、あまり抽象的に

なりすぎないように、AI・データサイエンスの技術手法を（数式などはできるだけ省略しながらも）具体的に解説するような構成としています。

まずは第１章と第２章にて、AI・データサイエンスの必要性や定義、ビジネス活用事例を紹介し、そして第３章以降で、AI・データサイエンスにおいて重要な技術手法の紹介をします。時間がない経営者の方などは、第１章と第２章だけでも学びになると思います。

第３章以降では、ビジネスパーソンがイメージしやすいような事例を取り上げて、その課題解決の方法として AI・データサイエンスの必要性を掲げます。そこで活用される技術手法の基礎知識を学び、かつ Excel を用いたハンズオン演習でデータを簡単に触っていただくことで、より具体的なビジネス適用イメージが湧き、腑に落ちやすいようにしてあります。

なお、本書を手軽に読んでいただきたいという思いもあり、手元で Excel を開かなくとも読めるように図解してあるので、しっかり腰を据えて学びたい方は Excel も使いながら、手軽に読みたい方は本書だけを読み進めて、と、どちらの形でも楽しめて、学べる内容になっています。

一方で実際にデータサイエンティストや機械学習エンジニアを目指したいといった方々は、本書の内容だけでは事足りない部分もあるので、本書でどうビジネスで活用できるかをまずは理解しましょう。そのうえで実装のための Python や機械学習のための数学、といった技術的な内容をキャッチアップするとよいと思います。それでは、一緒に学んでいきましょう。

三好大悟

CONTENTS

練習用ファイルのダウンロードについて

実践パートで使える練習用ファイル（xlsx 形式）は、以下の URL からダウンロードできます。

https://book.impress.co.jp/books/1121101015

※画面の指示に従って操作してください。
※ダウンロードには「CLUB Impress」への登録が必要です（登録は無料）。
※練習用ファイルは、本書籍の範囲を超えての使用を想定していません。

Chapter 1

データサイエンスを
ビジネスで活用する

Section

01 なぜいまデータサイエンスの必要性が叫ばれているのか？

最近スキルアップを目論んで本屋のビジネス書コーナーによく行くんですけど、文系のための AI とか、データサイエンス人材というキーワードが気になるんです。

たしかに近頃は一般のビジネス現場でも、AI やデータサイエンススキルを持っているといろいろと役立つ場面が増えてるわね。

でも僕らみたいな小さいマーケティング会社で、AI やデータサイエンスの知識を使う機会ってあるんですか？

もちろんあるわよ。マーケティングなんてデータを活用してなんぼの世界じゃない？ データサイエンスはマーケティングだけではなく、あらゆる業種や職種において役立つ非常に強い武器といっても過言ではないわ。

先輩とても詳しそうですね。ぼくもスキルアップのためにぜひ身につけたいです！ 弟子入りするのでゼロから教えてください。

本書で学ぶデータサイエンスの範囲

- ☑ データの集計、可視化、記述統計
- ☑ 機械学習（教師あり学習、教師なし学習）
- ☑ レコメンデーション、数理最適化

「データサイエンス」とは？

突然ですが、皆さんは「データサイエンス」と聞いて、どのようなイメージを持ちますか？ まったくイメージが沸かない人もいるでしょうし、人それぞれ持っているイメージは異なっていると思います。営業を主な業務としている人のことを「営業担当」「営業マン」などと呼ぶように、データサイエンスを主な活動生業としている人を「データサイエンティスト」と呼びます。しかし「営業マン」に比べると、「データサイエンティスト」がどんな業務をしているのかイメージできる人は少ないのではないでしょうか？

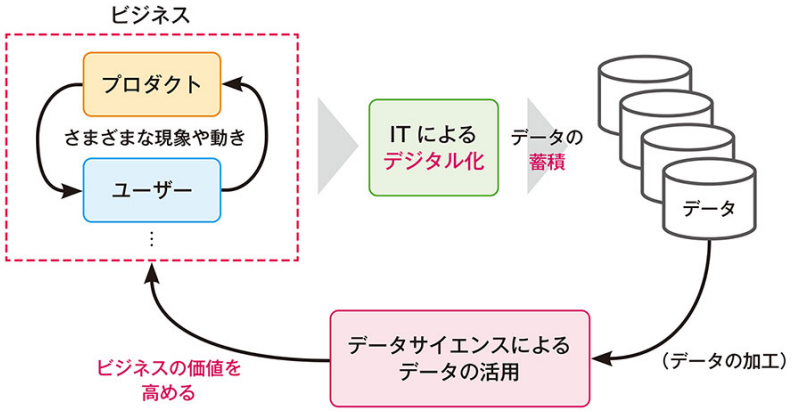
データサイエンティストはいわゆる専門職ですが、同じ専門職である医師や弁護士などのように、明確な資格があるわけではありません。また同じ「データサイエンティスト」でもまったく異なる業務をしていることはよくあります。その前提のもとで、本書では「ビジネスにおけるデータサイエンス」という位置づけで話を進めていきます。その文脈において、私なりにデータサイエンスをわかりやすく定義してみると、**「ビジネスにおけるデータを活用し、そのビジネスの価値を高める営み」**と考えることができます※1。もう少し詳しくみていきましょう。

そもそもデータはどこからともなく湧き出てくるものではありません。特にビジネスにおいては、そのビジネスのプロダクトやユーザー（顧客）、あるいはオペレーションなどを中心に、さまざまな「モノ」や「ヒト」の現象や動きがあり、それをITによりデータ化（デジタル化）しているにすぎません。

そしてそのデータを蓄積し、必要に応じて加工し、何かしらの形で活用することにより、そのビジネスに対して意味あるインパクトを与えることが、データサイエンスのなすべきことと考えられます（具体的なデータの活用方法は第2章以降で紹介していきます）。

（※1）もちろん人によって考えている定義は異なるので、あくまで一個人の意見として受け止めてください。

➲ データサイエンスによりビジネスの価値を高めるサイクル［図1-1-1］

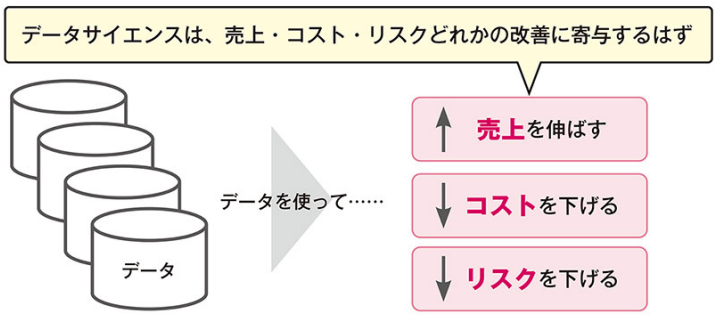


「ビジネスの価値を高める」というフレーズは少々抽象的ですが、ここでは、次のような行為を指しています※2。

・売上に関する指標（顧客単価など）を向上させる
・何かしらのコストやリスクを削減、低減させる

もちろんデータサイエンスだけではなく、営業やマーケティング、経理といったさまざまな業務が上記のようなことを達成するために必要とされています。ことデータサイエンティストは、「データを（武器として）活用する」ことで、そのような貢献を求められる、ということになります。

➲ データサイエンスはデータを使って売上、コスト、リスクを改善する［図1-1-2］



※リスクは広義にはコストに含まれていると考えられます。

（※2）あるいは直接的には見えにくいですが、競合他社にはない機能・差別化要素を生み出すといった価値の出し方もありえます（もちろん将来的には利益に貢献するはずです）。

なぜいまデータサイエンスが必要なのか？

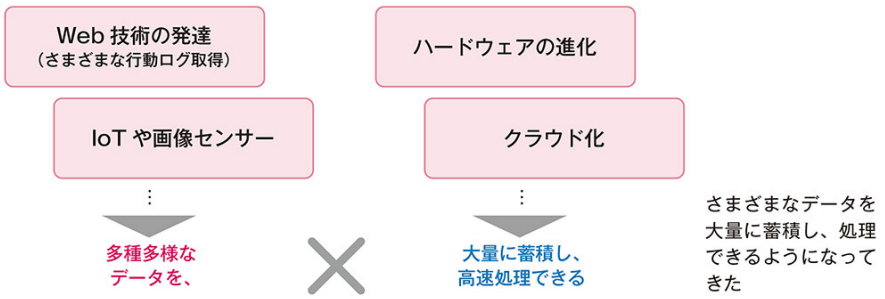
それでは、なぜいまデータサイエンスが必要なのでしょうか？ それは、デジタル化によるデータの蓄積が進んできているということに加えて、2つの点が挙げられます。

1つは、WebやIoT、画像センサーといった技術の発達により、行動ログなどの詳細なユーザー情報、テキストや画像といった、**非常に多種多様なデータを蓄積できるようになってきている点**です。これにより、古典的なユーザーの属性情報（性年代や住所など）といったものだけではなく、より多くの現実の事象をデータとして取得することができるようになり、データ活用の幅が広がってきています。

2つ目としては、メモリやCPU、GPUといったハードウェアの進化や、クラウド化が挙げられます。それらの恩恵により、自社にしっかりとしたデータセンターがなくとも、**大量のデータを蓄積しやすくなり、かつ大量のデータを高速に処理しやすくなってきています。**今までは多種多様なデータを大量に保存しづらいという問題や、保存できたとしても処理に時間がかかりすぎて実務的ではない、といった課題がありました。しかし上述の恩恵により、それらが克服されてきています。

これらの観点から、「**多種多様なデータを大量に蓄積することができ、かつ高度な技術を高速に処理できるようになった**」ため、データサイエンスにより価値を出しやすくなってきているということです。そしてこれらの進化は指数関数的に進んでいるので、データサイエンスの必要性はこれからも高まっていくでしょう。

➲ 高まり続けるデータサイエンスの重要性［図1-1-3］



このことで、ほかにも変化が起こりえます。それは、これまでデータの活用は、デジタル化によるデータの蓄積がしやすかったIT・ソフトウェア業界を中心に進んできていましたが、さまざまな業界でデータの取得と活用が進んでいくということです。

これまでのIT・ソフトウェア業界におけるデータの活用例としては、ほんの一例ですが以下のようなものが挙げられます。

・ECサイトにおいて、ユーザーに最適な商品を推薦（レコメンド）し、購買金額を向上させる
・広告出稿ビジネスにおいて、広告に興味を持ってくれそうなユーザーを特定し、コンバージョン（CV）率を向上させる
・出品サイトにおいて、出品時の最適価格を提示し、売買のマッチング率を向上させる

IT・ソフトウェア以外の業界においても、次のようなデータの活用例が挙げられます。

・小売業界で、ユーザーの店舗動向をデータ化し販売促進に活用する
・物流・配送業界で、配送情報をデータ化して配送の効率化を図る
・製造業界で、製品の画像情報をデータ化して不良検知を自動化する

つまり、あらゆる業界で、さまざまなデータを取得し、活用する契機が起こり始めているということです。そのため私は、営業やマーケティングといった職種と同様に、データサイエンティスト（職種名が将来どうなるかは未知ですが……）も、すべての業界、そして会社に配置されるべき職種になっていくのではないかと考えています。

実際に、「データアナリスト」「データサイエンティスト」「AIエンジニア」「機械学習エンジニア」など職種名はさまざまですが、データ活用のスペシャリストである職種の求人数は、幅広い業界で増えてきています。

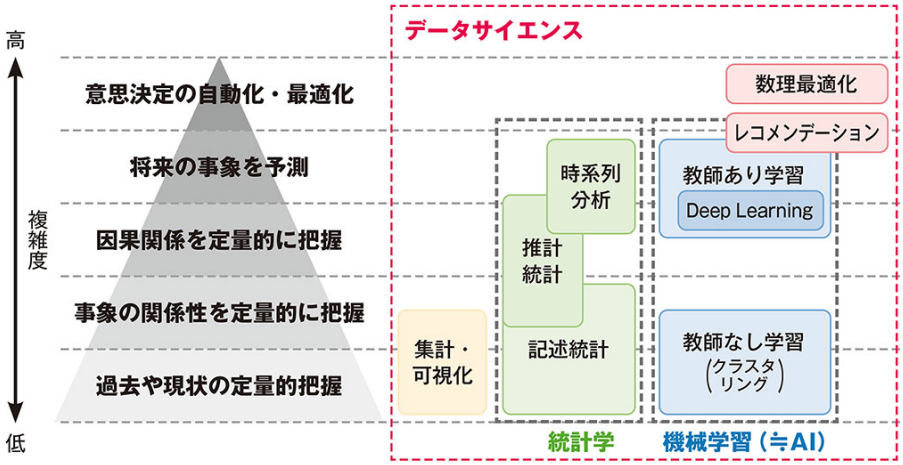
Section

02 AI やデータサイエンスにおける技術概観

AIやデータサイエンスの守備範囲

さて、これまでは細かい説明なしで「データサイエンス」という単語を用いていましたが、皆さんは「AI」という言葉のほうが聞き慣れているかもしれません。どちらも似たような単語ではありますが、本書では統一して「データサイエンス」という言葉を使用していきます。AI やデータサイエンスの定義に関して、（実際、明確な定義があるわけではないですが）次の［図1-2-1］をもとに整理します。データサイエンスと AI の守備範囲は、それぞれがどういった技術を主に活用しているかが異なっていると考えられます※3。

➲ データサイエンスや AI などの守備範囲［図1-2-1］



※あくまでイメージです。実際は多種多様な手法や分野に分かれています

［図 1-2-1］で紹介した各技術トピックをもう少し紐解いていくと、以下のような役割分担があると考えられます。

（※ 3）前提として、データサイエンスや AI の定義は、人や場合により異なることがよくあります。また図の各技術の定義や境界線は、初学者でもとっかかりやすいように表現している部分もあります。あくまでビジネス向けのわかりやすさを第一優先としており、学術的な文脈ではより細かい補足が必要という点はご容赦ください。

各技術分野の概要

＜集計・可視化＞

集計・可視化とは、これまでの**過去データを集計して、過去の傾向を可視化し、把握する**といった役割を担います。Excel などで行うグラフ化などもこれにあたります。統計解析と似ている部分もありますが、集計や可視化は、あくまで**手元のデータの傾向のみを読み解いている**だけですが、統計解析では手元にないデータの傾向も鑑みて読み解く、という違いがあります。

＜統計学＞

機械学習と被る部分もたぶんにありますが、主に、**少量のデータであっても**、データの傾向から**その元となる集合全体**※4**の現象を数式で記述**し、**現実世界の傾向をできるだけ正確に読み解く役割**を担います。また統計学と一言でいっても、以下のようなさまざまな分野に分かれます※5。

- 記述統計：平均や標準偏差といった手元のデータを集計する方法論
- 推計統計：手元のデータから、それらを含んだ集合全体のデータを読み解く方法論
- 時系列分析：（推計統計にも含まれますが）時系列の要素を加味し、将来情報を読み解く方法論

＜機械学習・レコメンデーション＞

機械学習は、**大量のデータ（ビッグデータ）をインプットし、時に高性能なコンピューティングパワーを利用して、未知なる現象を予測する**役割を担います。昨今よく耳にする**「ディープラーニング」という高度技術も、機械学習の手法のうちの１つ**です。

また、機械学習には大きく**「教師なし学習」と「教師あり学習」**があり、教師なし学習は集計・可視化や統計学のように、過去データの関係性の定量把握が主な役割で、教師あり学習はより将来事象の予測に重きをおきます。

レコメンデーションとは、**ユーザーの好み（興味関心があるであろう）と思われる情報を提案する**ようなソリューションです。レコメンデーションエ

（※4）そのような手元のデータの背後にあるであろう、（取得することはできないであろうが）すべてのデータのことを「母集団」といいます。
（※5）学術的にはより細かく分かれています。

ンジンはビジネス活用が特に盛んで、またレコメンデーションに特化した技術も多いので、［図 1-2-1］では少々くくり出して表現していますが、機械学習の技術などを使いながらエンジンが開発されることも多いです。

＜最適化＞

対象となる変数※6を変えたときにターゲットとする目標値がどう変化するかを探索することで、**どのような変数の値にすれば、目標値が最大化／最小化するかを算出する役割**を担います。こちらも後ほど具体的に紹介します。

まずは、それぞれ（重複する部分もありつつ）役割が異なっている技術トピックである、ということを理解できれば大丈夫です。

世間一般にいわれている「AI」は、主に機械学習を活用した技術を指していることが多く※7、それらの技術で「**特に人間の行動傾向を把握し、再現する**」といったことを目指しています。囲碁や将棋でプロにも勝るようなAIアルゴリズムを開発したり、ロボットにAIアルゴリズムを搭載して人間同様の動きをさせたり、などが典型的な例です。

一方で「データサイエンス」は、［図 1-2-1］でマッピングした技術活用すべてを指します。つまり、データサイエンスというのはデータを活用する営みすべてを指し、より「ビジネスにおけるデータを活用し、そのビジネスの価値を高める営み」に近い表現であると考えられます。AIも上記の表現にかなり近い言葉としてよく使われますが、もう少し**「高度な技術や処理能力を活用する」といったニュアンスが強い**と感じます。したがって本書では、**「データサイエンス」という言葉を使用していきます**。

本書で取り上げる技術分野

なお本書では、私の経験やこれまでの社会実装の状況などを鑑みて、次ページの［図 1-2-2］の赤枠で示している通り、**主に機械学習、加えてレコメンデーションと数理最適化**を重点的に取り扱います。これらの技術は、ここ近年で急速にビジネス実装が増えてきたものです。ただし技術的に難しい手法は、（ある程度ビジネス活用されているものでも）今回は紙幅の関係上取り扱わ

（※6）変数とは「変化する値」を指します。たとえば、商品単価というのはずっと一定ではなく状況によって変えていく値になりうるので、変数であると考えられます。
（※7）曖昧な部分もあり、統計解析を使用したものも含まれることがあります。また集計（ルールベースの処理）がAIの範疇に含められることもあります。

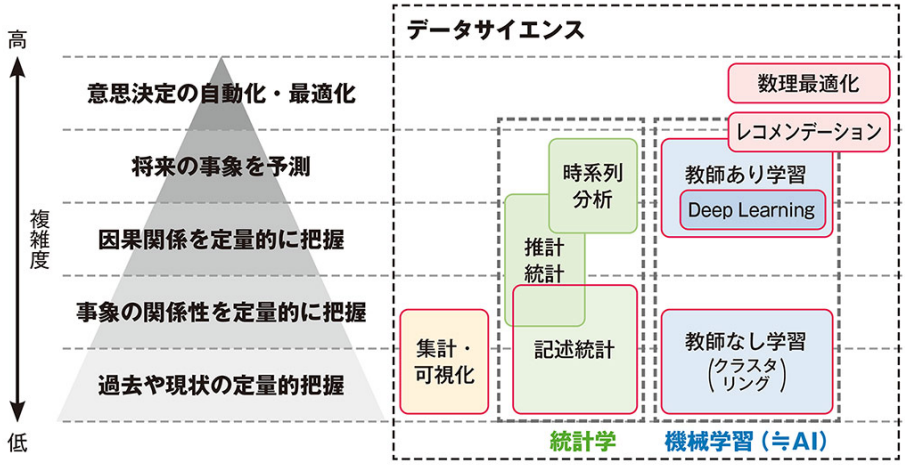
ないこととします。

一方で、データの集計・可視化、基礎的な統計解析は、データを適切に理解するために必要な技術です。機械学習などの高度技術といえども、（何より“データ”サイエンスというくらいなので）データの理解は欠かせません。そこで第3章で、非常に基本的な記述統計とデータの可視化を取り上げます。特に簡単な記述統計（集計）や可視化は、普段Excelを扱っていれば比較的イメージのつきやすい分野だと思います。第3章で重要な点を抽出・要約する形で進めていきましょう。そのうえで、第4章以降で機械学習などの技術を学んでいきます。

なお、推計統計や時系列分析は比較的学術的な説明が多いので、今回は取り上げません。実務的によく使用する仮説検定や回帰分析といったトピックに関しては、手前味噌で申し訳ないですが、私が以前に執筆した『統計学の基礎から学ぶExcelデータ分析の全知識』（インプレス刊）を参照していただければ、その考え方や、実務的な分析方法の理解が進むと思います。

最近では、「AI」というワードの認知度が広がりすぎてしまっているために、本来はデータサイエンスというべきところを、コミュニケーションを円滑にするために（ぐっとこらえて）「AI」といってしまうようなケースも少なくないわね。現場や業務の中で、特にデータサイエンスに関して明るくない方と接する際は、ある程度の使い分けの不明瞭さは許容してしまっても大丈夫よ。

➲ 本書で主に取り上げる技術（赤い枠の部分）［図1-2-2］



※あくまでイメージです。実際は多種多様な手法や分野に分かれています

Chapter 2

データサイエンスの手法を理解する

Section

01 データサイエンスの手法ごとの特徴をつかもう

データサイエンスやAIというものがいくつかの技術分野で成り立っていることはわかりました。

データサイエンス自体はとても幅広い領域が含まれるんだけど、ひとまずは機械学習やレコメンデーション、数理最適化などの技術がどんなもので、どう役立つかを理解すれば大丈夫よ。

実は、データサイエンスが「どんなものか」「どう役立つか」の部分がまだイメージできなくて……。

それはこれから学んでいくから慌てなくても大丈夫。特にビジネスでは、データサイエンス技術は「どう役立つか」がわかってはじめて意味を持つものだから、新しい技術を学ぶときに、「どういう仕組みか」「ビジネスにどう役立つか」を考えるのはとてもよいことよ。特に「技術屋」ではない私たちの仕事はまさにそこね！

基本的な仕組みを理解し、理解することでビジネスにどう役立てられるかを考えるべし、ということですね。俄然やる気が湧いてきました！

ここで学ぶこと

- ☑ データサイエンスの主要技術を理解する
- ☑ 各技術がどういう仕組みかを考える
- ☑ 各技術がビジネスにどう役立つかを考える

本書で扱うデータサイエンス手法の概観

前の第１章でデータサイエンスの定義や守備範囲を把握したうえで、もう少し具体的に、本書で取り扱うデータサイエンス手法を紹介していきます。また随所にビジネスでの活用事例なども折り込んでいくので、イメージを膨らませていきましょう。

ここでは、本書で学ぶ以下の技術手法の概要を紹介していきます。そして各章で、どのような技術なのか？ ビジネスシーンでどう活用すればよいか？といった具体的な内容に関して、ビジネスケースをベースに、しっかりと学んでいきましょう 。

- 集計、可視化、記述統計 ……第３章
- 教師あり学習（回帰問題）……第４章
- 教師あり学習（分類問題）……第５章
- ディープラーニングによる画像解析……第６章
- 教師なし学習 ……第７章
- レコメンデーション ……第８章
- 最適化 ……第９章

次 Section 以降で、各章で学ぶ技術の概要を簡単に紹介します。繰り返しになりますが、詳細は各章で学ぶので、ここではざっくりとどのような技術なのか、といった点を押さえられれば大丈夫です。なお、第３章で取り扱うデータの集計・可視化・記述統計（平均値や中央値など）は、技術手法の概要はそこまで難しくなく、かつ活用事例は非常に幅広くなってしまうので、次 Section 以降では特に紹介しません。

本来、ディープラーニングも教師あり学習の一種であったり、また教師あり学習を利用したレコメンデーションエンジンなども存在していたりするので、技術的な観点からは、今回の章立ては正確ではない部分もあるの。でもここでは本書の狙いである「どうビジネスで活用するか」という観点で頭が整理できるように、ビジネス活用方法をベースに７通りに分類しているのよ。

Section

02 教師あり学習（回帰問題・分類問題）

ビジネスにおけるデータサイエンスとして、最もよく使われている技術の1つが、この「教師あり学習」です。回帰問題と分類問題で異なる部分が少々ありますが、どちらも「教師あり学習」なので、まずは教師あり学習の概要をざっくり把握しておきましょう。

教師あり学習の概要

教師あり学習とは、その名の通り「教師」となるデータを学習することが大きな目的となります。たとえば、あるサービスを運営する会社が満足度の低い顧客に対して、ロイヤルティを少しでも高めてもらうための施策を打ちたいとしましょう。その際に、「満足度の低い顧客＝サービスを解約してしまいそうな顧客」と読み替えれば、解約しそうな顧客を特定することで、その顧客群に対して施策を打てます（当たり前ですが、満足度の高い顧客にわざわざその施策を打つ必要はありませんよね）。

となると、その「解約してしまいそうな顧客」をいかに正確に特定するかが重要になります。ここで「教師あり学習」の登場です。過去のデータを収集して、どのような顧客が解約したか、または継続しているか※1、という学習データを用意します。

- 「年齢・性別・過去1週間の閲覧回数……」などといった顧客の情報※2のインプットデータ
- その顧客が「解約したか（1）・継続しているか（0）」というアウトプットデータ＝（インプットデータの）教師となるデータ

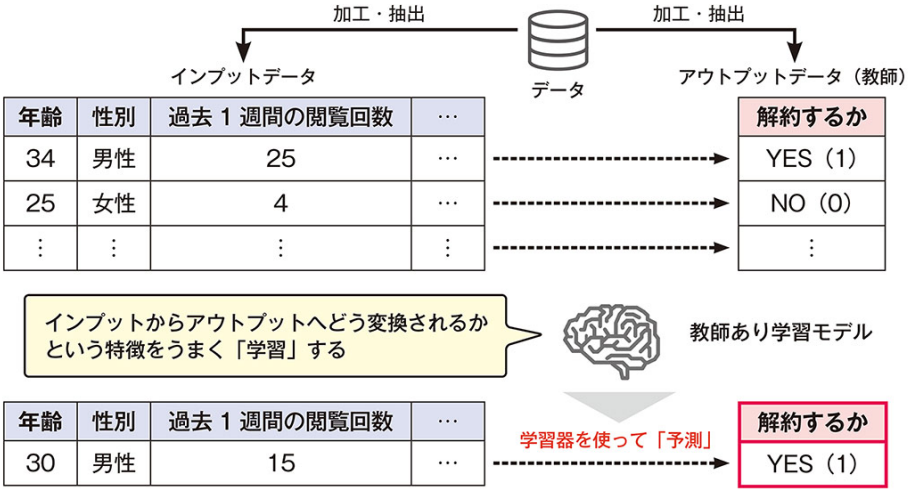
そして「どのようなインプットのときにどのようなアウトプットになるか」を教師あり学習モデルに学習させます（世の中では、この教師あり学習モデ

（※1）正確には「ある時点を起点に、将来X か月後に解約しているか・継続しているか」といった判定基準を設けて、過去のデータを集計することとなります。
（※2）このような情報を「特徴量」といいます。次の第3章以降で詳しく紹介します。

ルがしばしば「AI」と呼ばれています）。

すると、この教師あり学習モデルが「どういう顧客であれば解約しそうか／継続しそうか」という**インプットデータからアウトプットデータへの変換の特徴を学習**してくれます。そうなることで、今の顧客に関して「年齢が30歳で、性別が男性で、過去1週間の閲覧回数が15回で……、という顧客は解約しそう」ということを「予測」（推論）できるようになります。

教師あり学習の全体像［図2-2-1］



その予測を今の顧客群に対して適用することで「解約しそうな顧客群」「継続しそうな顧客群」を判別できるようになります。これで解約しそうな顧客群に対して適切に施策を打つことができるでしょう。「どのようなインプットデータを入れるべきか？」「教師あり学習モデルとは具体的にどのようなものなのか？」「予測した際の精度はどう判断すればよいのか？」といった具体的な話は、第5章以降で詳しく学んでいきます。

Tips　モデルとは？

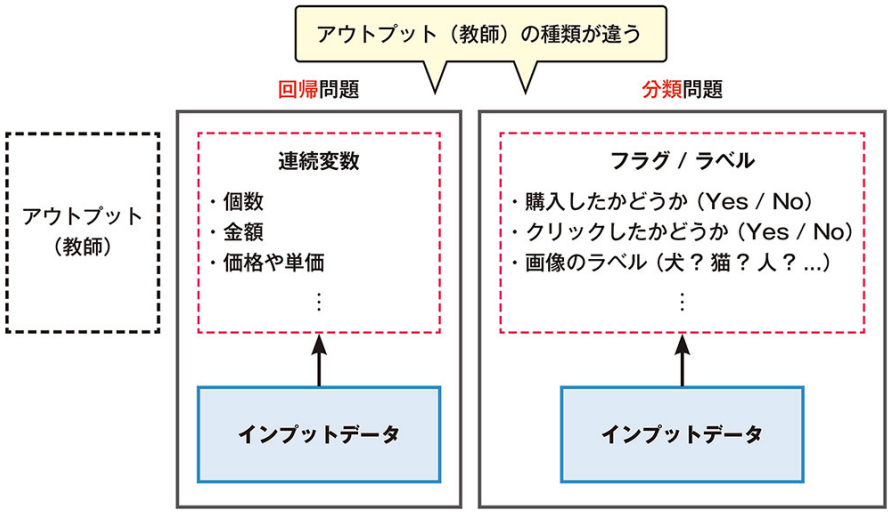
「モデル」には「模型」といった意味合いがあります。すべてのデータを理解するのは不可能なので、さまざまなインプットデータがどうアウトプットデータへ変換されるかという傾向を、モデルという型に落とし込むのです。そのモデル（型）をどう定義するか、というモデルの種類に関しては第4章以降で詳しく学びます。

回帰問題と分類問題の違い

さて、第 4 章は回帰問題、第 5 章は分類問題という形で分かれていますが、どちらも教師あり学習です。では両者の違いはなんでしょう。それはアウトプットとなる教師データの種類です。

➲ 回帰問題か分類問題かでアウトプット（教師）の種類が違う［図2-2-2］

- 回帰問題：教師データが「連続変数」となり、個数・金額・価格といった値を学習する際に使われる
- 分類問題：教師データが「フラグやラベル」となり、購入したかどうか (Yes / No)・クリックしたかどうか (Yes / No)・画像ラベル（犬・猫・人……）といった値を学習する際に使われる



回帰問題･分類問題それぞれにおける、具体的なビジネス活用例としては、以下のような例が挙げられます。

＜回帰問題＞

・不動産物件情報から、その不動産の価格を学習する

・商品の販売情報から、その商品の販売個数を学習する
・出稿した広告予算情報から、広告経由での売上金額を学習する
・ユーザー（顧客）の購入履歴情報から、そのユーザーの購入金額を学習する

＜分類問題＞

・これまでのユーザーの行動履歴情報から、ユーザーが解約してしまうかどうかを学習する
・過去の購売情報から、あるユーザーがある商品を購入してくれるかどうかを学習する
・画像データから、その画像のカテゴリである画像ラベル（犬・猫・人など）を学習する
・日本語の文章データから、そのテキストの記事カテゴリ（経済・スポーツ・エンタメ……）を学習する

そのほかにもさまざまな活用方法が考えられますが、まずは教師あり学習には主にこの２種類があり、アウトプットの種類によって区別される、ということを押さえておきましょう。

とはいえ教師データの種類が異なるだけなので、先ほど紹介したような教師あり学習としての本質やロジックは変わりません。

ただし、回帰問題と分類問題とで「どのような種類のモデルを使用すればよいか？」「その学習モデルの精度をどういった指標で判断すればよいか？」といった点が異なります。そのため、両者をしっかりと区別できるようにしておく必要があります。具体的にどう異なってくるのかは第４章と第５章で学習します。最終的に、ビジネス上でどういう課題の際には回帰問題または分類問題を使用して、どのようなモデルを構築して、どのように精度を確認すればよいか？といった形で理解ができるようにしましょう。

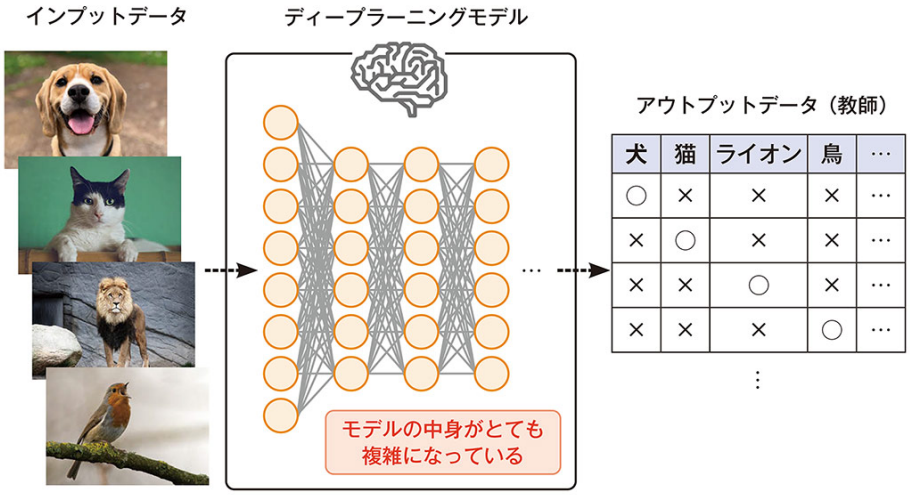
Section

03 ディープラーニングによる画像解析

画像解析とは？

続いて画像の話題に移りましょう。一口に画像解析といっても、その方法は非常に多岐に渡ります。そこで本書では、一番基本的かつ重要な「**画像分類**」を取り上げます。問題設定としては、「**その画像はどのカテゴリ（画像種類）に分類されるか**」を大量の画像データから学習させ、新たな画像がきた際にその画像はどのカテゴリの画像なのかを、精度よく予測（推論）させることがゴールとなります。

➲ 画像データから、複雑なネットワークモデル通じて、画像の種類を分類［図2-3-1］

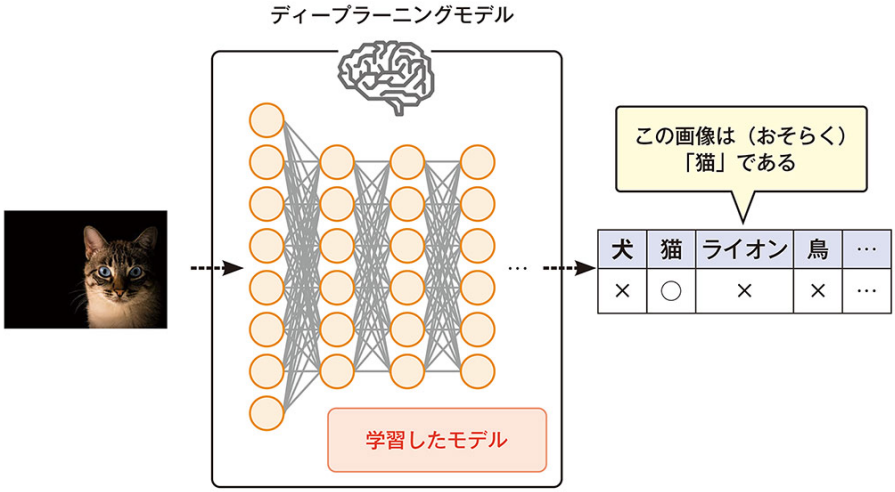


まずは対象とする画像を決め、それらをどう分類するかという「カテゴリ」を定義し、画像ごとにラベリングします。たとえばこの画像は「犬」、この画像は「猫」、といった具合です。仮に画像のカテゴリが「犬・猫・ライオン・鳥……」で100種類ということとなれば、それは100カテゴリの分類問題と

いうことです。

そしてある画像がどの画像カテゴリかを、ディープラーニングモデルに学習させます。その結果、ある新たな画像に関して、その画像は何のカテゴリなのかを予測（推論）できるようになります。

➲ **学習したディープラーニングモデルを使って、画像から種類を予測［図2-3-2］**



少し話が逸れますが「画像分類」というのは、先ほど学んだ教師あり学習の分類問題と同じものです。つまり**「インプットデータとして画像・アウトプット（教師）データとしてラベル」が用意された教師あり学習**（の分類問題）であるといえます。

それでは何が特殊かというと、そもそも**インプットデータが画像である**という点に加えて、インプットデータの画像からアウトプットであるラベルに変換するために「ディープラーニング」という複雑なモデルを使用する必要がある、ということです。詳しくは第6章で学習します。

なお、教師あり学習の章で解説してもよいのですが、ビジネス活用という観点で画像を取り扱うことは少し特殊なケースなので切り離しています。第4章と第5章の教師あり学習では、皆さんが普段Excelで取り扱うような、行と列があるデータ（「構造化データ」といいます）を対象とします。

さまざまな画像解析による活用例

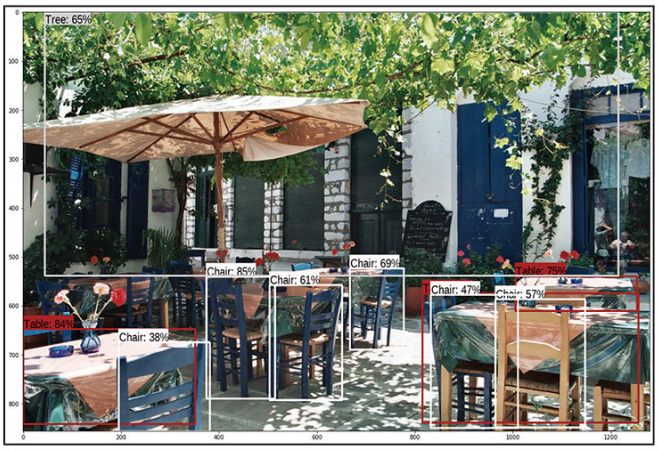
まず「画像分類」をしっかり理解できれば十分ですが、ビジネスシーンでは画像分類以外の技術もしっかりと活用が進んできています。ここで簡単にいくつか事例を紹介しましょう。

❶ Object Detection（物体検出）

1つはObject Detection（物体検出）です。画像分類は「1枚の画像に関して、それがどのカテゴリの画像なのか？」を考えるモデルでしたが、物体検出は「**ある画像に関して、どこの座標に、どのような物体が存在しているか**」を考えます。

物体がある部分に関して、その座標情報を学習し、かつそこにどのような物体があるか（木か？椅子か？机か？……）までを判定します。

➲ 物体検出（Object Detection）のイメージ［図2-3-3］



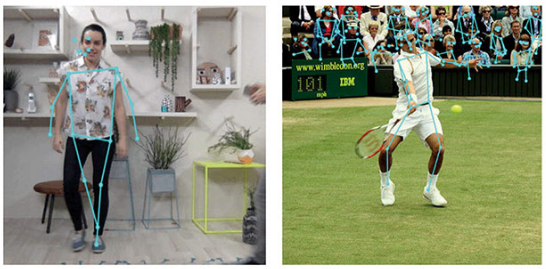
画像のどこ（座標）にどのような物体が存在しているかを検出
出典：https://www.tensorflow.org/hub/tutorials/object_detection

たとえば店舗内に設置したカメラで、どこに人間がいるかを検出できれば、「どのような時間帯に、どのくらいのユーザーが店内にいるかを把握できる」「ユーザーがどのような導線で店舗内を歩き回っているかを把握できる」ようになります。そのようなデータを用いて販促活動などに活用できます。

❷ Pose Estimation（姿勢推定）

2つ目はPose Estimation（姿勢推定）です。この技術は、画像内の特に人物に着目して、「**画像内の人物において、どこの座標にどのような関節点があるかを検出**」するような問題設定となります。

➲ **姿勢推定のイメージ［図2-3-4］**



画像の人体において、どこ（座標）にどのような関節点が存在しているかを検出
出典：https://www.tensorflow.org/lite/examples/pose_estimation/overview

活用方法としては「スポーツなどで、どのような動きをしているかをより詳細に把握し、適切なアドバイスができる」「店舗や家の前などに設置したカメラを用いて、人物の関節点の動きから、万引や侵入などの防犯検出ができる」といった内容が考えられます。

物体検出も姿勢推定も、それ単体で売上やコストを改善するというよりは、データ収集の1つの手段として用いられることが多いです。今までは取得できなかった（あるいは取得しようとすると膨大なコストがかかる）データを用いて、より精度の高い業務改善や販売促進などの施策ができるようになっています。

❸ Style Transfer（画風変換）

最後はStyle Transfer（画風変換）です。これは文字通り「**ある画像のスタイル（画風）を変換して、新たな合成画像を生成する**」という問題設定です。さまざまな画風へ変換できることにより、「画像の鮮明度をよくする」「アニメ風の画像や絵画風の画像を手に入れることができる」といった活用方

法があります。画風の変換だけではなく、画像そのものを生成する（Image Generation）といった技術も進歩してきており、比較的容易に画像を取得できるようになってきています。

なお、そのようなディープラーニングにより生成された画像の著作権はどこに帰属するのか？ といった法律的な問題も出てきています。まだ法整備も追いついていない状況であるため、今後どのような法体系となってくるのか、といった点も注目です。

➲ 画像変換のイメージ［図2-3-5］

画像のスタイル（画風）を変換して、新たな合成画像を生成
出典：https://www.tensorflow.org/lite/examples/style_transfer/overview

Tips ディープラーニングモデルの使いどころ

正確を期すために補足すると、ディープラーニングのモデル自体は、必ずしも画像データだけに特化しているわけではなく、どのようなデータに対しても適用可能です。ただし、こと画像データを扱うときは、（少なくとも現時点では）ディープラーニングモデル一択という状況です。

- さまざまなデータ
  →ディープラーニング含めていろいろなモデルを適用可能
- 画像データ
  →ディープラーニングモデルを適用

と理解するのがよいでしょう。

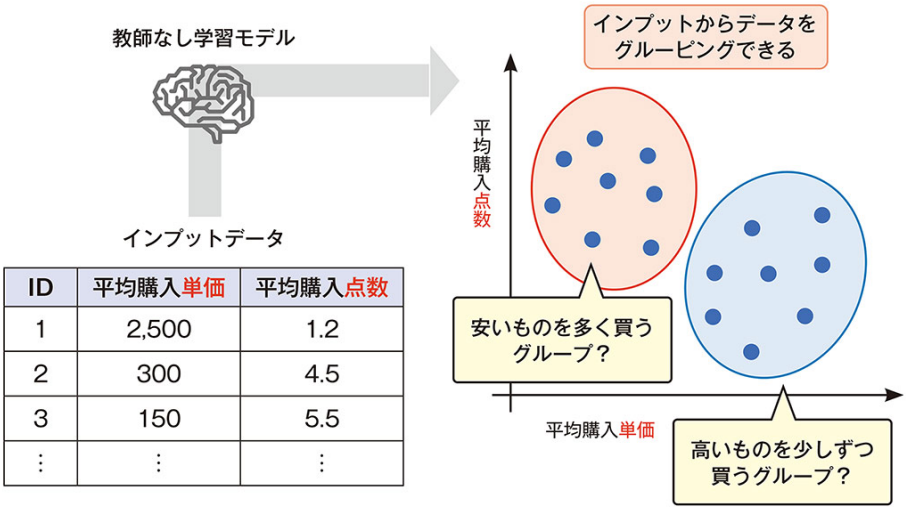
Section

04 教師なし学習

教師なし学習の概要

続いては「教師なし学習」です。教師「あり」学習と異なり、教師「なし」なので、その名の通り**教師データが存在しません**。インプットデータのみとなります。ではインプットデータのみで何をするのかというと、平たくいえば、「データをグルーピングする」役割を果たします。［図 2-4-1］を見てください。たとえば自社 EC サイトなどでユーザーごとのインプットデータとして平均購入単価や平均購入点数といったデータを用意します。その際に、その 2 つの情報（特徴量）をもとにユーザーデータをいくつかにグルーピング（よく「クラスタリング」と呼ばれます）できます。

➲ 教師なし学習の概要［図2-4-1］



ID	平均購入単価	平均購入点数
1	2,500	1.2
2	300	4.5
3	150	5.5
⋮	⋮	⋮

上図は 2 グループにグルーピングしている

［図 2-4-1］では 2グループに分けていますが、何グループにするかなど、できるだけ適切に設定する必要があります。

また、図を見て「これであれば単にデータを散布図にプロットすれば問題ないのでは？」と思ったかもしれません。この例では問題ないかもしれませんが、もし情報（特徴量）が 3つ、4つ以上あった場合どうしましょう？（年齢や性別、過去の閲覧回数などなど）。その場合は、とてもプロットすることはできないので、可視化だけでの判断が難しくなってきます。そういうケースで、教師なし学習によるグルーピングが有効になります。

教師なし学習の結果を解釈する

ただグルーピングしただけでは意味がなく、それぞれのグループがどのような特徴や傾向をもったグループなのかを**解釈**する必要があります。たとえば、次のような解釈です。

・グループ１は、安い商品を一度に多く買うグループなのではないか？
・グループ２は、高い商品を少しずつ買うグループなのではないか？

ここから、次のような施策に結びつけられるでしょう。

・グループ１には、もう１商品、お得な商品を買ってもらえるように、併買のためのポイントを付与しよう
・グループ２には、できるだけ購入商品を気にいってもらえるように、高級だが高評価な商品をメールで紹介してあげよう

このようにグループごとに最適な施策を打つことができます。そうすると、全体でみたときに、ビジネスをより改善させられます。実際にどのようにグループを解釈していけばよいか？ といった部分は、第７章で詳しく見ていきましょう。

Section

05 レコメンデーションの事例

レコメンデーションの代表的な事例

続いては「レコメンデーション」について簡単に紹介します。「レコメンデーション」という言葉自体はよく耳にすると思います。具体的な手法の前に、いくつか代表的な例を紹介しましょう。

真っ先に思いつくのは Amazon です。皆さんも一度は Amazon で商品を購入したことがあると思いますが、ある商品ページを開くと、その商品と「よく一緒に購入されている商品」が表示されています。また「この商品に関連する商品」も提示されます。

➲ おすすめ商品の例［図2-5-1］

Amazon で私の前著の本について、よく一緒に購入されている商品などが表示されている

これらは、これまでの Amazon 利用ユーザーの、購入履歴や Web 上の閲覧履歴、レーティング（評価）履歴などの膨大なデータに基づいて、ユーザーが一緒に購入してくれそうな商品を「推薦＝レコメンデーション」しているのです。仮にユーザーが「お、この商品もよさそうだな。ついでに購入して

おくか」と**一緒に購買してくれれば、「購買点数」や「1 購入あたりの顧客単価」が向上することで、Amazon の売上が向上**します。

Amazon はレコメンデーションの代表例としてよく紹介されますが、近年だと Netflix（ネットフリックス）もレコメンデーションシステムが非常に有名です。Netflix は定額制の動画ストリーミング配信サービスを提供する会社ですが、前身のオンラインでの DVD レンタルサービスの時代から、レコメンデーションのシステムを導入していたそうです。

さらに今の Netflix は、「アクション・アドベンチャー」「国内ドラマ」「ラブ・ロマンス」などのさまざまな軸で、オススメとなる動画を推薦してくれます。このおかげで、ユーザーもさまざまな軸で動画を探すことができそうです。

当然、ユーザーがより多くの動画を見てくれることで、継続的に定額課金を続けてくれる＝ユーザー LTV（Life Time Value。顧客生涯価値）が改善する、ということにつながります。

McKinsey が出した 2013 年時点の記事にも、**ユーザーが Amazon から購入する商品の 35%、Netflix で視聴される動画の 75% がレコメンデーションシステムに由来しているという調査結果**が載っています※3。レコメンデーションは、ユーザーとの接点に近く、売上の KPI（Key Performance Indicator。重要業績評価指標）に直結しやすい、かつユーザー体験（UX：User eXperience）の向上にも寄与しやすいので、ビジネス活用との相性が非常によいのです。

レコメンデーションエンジンの概要

さて、レコメンデーションのイメージが湧いてきたでしょうか。レコメンデーションとは、一言でいえば「**どれに価値があるかを特定するのを助ける道具**」であると考えられます。Amazon や Netflix の例からより具体的に、**ユーザーの好み（興味関心があるであろう）と思われる情報を提案する**と言い換えることができます。これにより結果的にユーザーの購買やコンバージョン（CV）※4 へつながっていきます。

それを実現させるために、レコメンデーションエンジンはどのような計算機能をもっているのでしょうか。詳細は第 8 章で紹介するので、ここではざっ

（※ 3）出典：https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers
（※ 4）Web サイトなどでユーザーが商品購入や問い合わせ、会員登録、契約など、利益に繋がる特定の行動をした状態

くりと見ていきましょう。レコメンデーションエンジンは主に、次の 2 通りのアプローチを行っています。

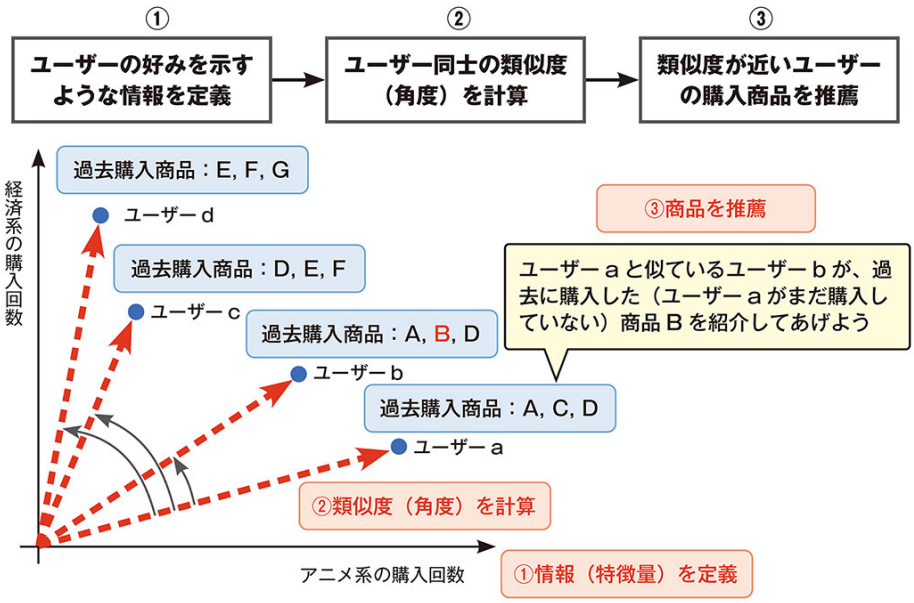
1. ユーザーベースのレコメンデーション
   → あるユーザーに似ているユーザー（が好むアイテム）を提案する
2. アイテムベースのレコメンデーション
   → あるアイテムに似ているアイテムを提案する

ベース（軸）はユーザーかアイテムかで異なりますが、どちらも本質的には同じことをやっていると考えられます。

ここではいったんユーザーベースの説明をします。手順としては、以下となります。

1. ユーザーの好みを示すような情報（特徴量）を定義
2. ユーザー同士の類似度（角度）を計算
3. 対象となるユーザーに関して、類似度が近いユーザーが過去に購入した商品（かつ対象ユーザーがまだ購入したことない商品）を推薦

➲ レコメンデーションシステムの概要（ユーザーベースの場合）[図2-5-2]



さまざまなポイントが潜んでいますが、まず重要なのは、**「ユーザーの好みを示すような情報（特徴量）は何か？（図のような２変数だけではなく）より多くの変数を考えて加える必要がある」**ということです。結局、計算は与えられた情報からしかできないので、「いかにユーザーの好みを示すであろう情報を与えられるか」ということがレコメンデーションエンジンの精度に寄与していきます。

そこで定義された情報をもとに、ユーザーごとに数値を計測します。それらは数学的に「ベクトル」と呼ばれる［図 2-5-2］の矢印のような状態で定義されます。つまり、**ユーザーごとにベクトルを持っているので、あとはベクトル同士の角度が類似度になる**、という算段です。

最後に、対象となるユーザーに関して、そのユーザーと類似度（角度）が近いユーザーを求め、そのユーザーが過去に購入した（もしくは高評価な）商品を推薦してあげればよいわけです。ただし、対象となるユーザーがすでに購入している商品を再度推薦してもあまり意味がないので、まだ対象ユーザーが購入していない商品を推薦する必要があります。

実際は、単に最も類似しているユーザーだけを見るのではなく、類似度の高いユーザーをより重視し、類似度がある程度近いユーザーもそれなりに重視し、推薦する商品を決めています（ユーザーに類似度のウエイトをかけているようなイメージです）。

本 Section の冒頭でも述べたように、レコメンデーションエンジンはビジネス活用が盛んであるため、それに伴って非常に速いスピードで技術発展している分野の１つです。

したがって、いま紹介したロジックは、あくまでシンプルなやり方の１つに過ぎず、前の Section で述べたような教師あり学習、ディープラーニングなどをレコメンデーションに応用する例も近年は増えています。その背景には、昨今のデジタル化やハードウェアの進化によってユーザーのさまざまなデータを大量に蓄積し、それらを高速に処理できるようになってきたこと、そしてビッグデータや複雑なアルゴリズムを活用できるようになってきたことが挙げられます。

少し話が逸れましたが、そういった背景もあることから、あくまで基本的な考え方として、これまでの内容を押さえておいてください。

Section

06 最適化

最適化の概要

最後に「最適化」を紹介します。まずは本書で学ぶ「最適化」とは何を指しているのか、その概念を押さえましょう。ビジネスのみならずさまざまな場面で最適化という言葉を聞いたことがあると思いますが、学問的には**「数理最適化」**と呼ばれます。私の経験上、最適化という言葉の定義が不明確なまま使われていることが少なくないと感じています。

たとえば「広告を最適化する」「人員配置を最適化する」などと使われていても、日常的にはそこまで気に留めることはありません。しかし数学的には「（数理）最適化」という学問には、明確な言葉の定義、使い方の定義が存在します。数理最適化とは、一言で表すと、**「ある対象となる変数をいろいろと動かしていく中で、変数によって動く決められた目的関数を最大化（あるいは最小化）すること」**を指します。イメージが沸きにくいので、具体例で見ていきましょう。

たとえば、何かしらの商品を売るときには、その商品の単価を決めるはずです。その際に、「売上（＝単価×販売個数）が最大となるような商品単価を決めたい」としましょう。そうすると、もちろんケースバイケースであることは百も承知ですが、一般的に次のような傾向があります。

・単価を下げすぎると、（当然ですが）売上単価が下がる
・単価を上げすぎると、今度は販売個数が下がる

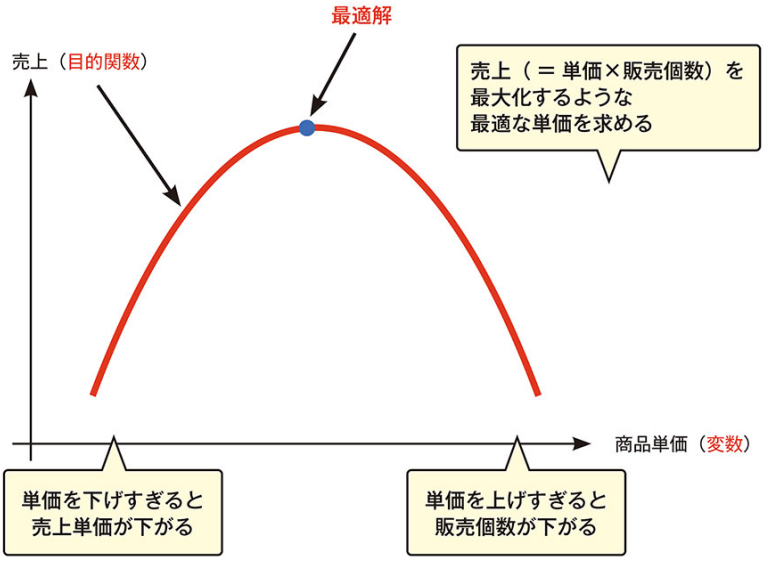
そのイメージを［図 2-6-1］に表しています。このことから、「高すぎず低すぎない単価」を求める必要がありそうです。つまり、もし単価と売上の関係性を何かしらの形で推測できれば、売上が最大化されるような単価を求めることができます。このことは次のように言い換えられます。

・「変数」（今回の例では単価）と呼ばれるレバーを動かすことで、
・「目的関数」（今回の例では単価によって変動する売上）と呼ばれる数式を最大化 / 最小化する

また、売上が最大化されるような単価を「最適解」と呼びます。つまり、今回の例でいえば、**「目的関数である売上を最大化するような、変数である単価の最適解を求める」**という問題設定であると考えられます。実際にどう最適解を求めていくのかは第 9 章で学んでいきましょう。

また多くの場合、「単価はあまりにも高すぎると受け入れらないので 10,000 円以下にしたい」といった制約もあるはずです。そのような変数に対してかける制約を「制約条件」といいます。こちらも詳しくは、第 9 章で見ていきます。

➲ **最適化の具体例：商品単価を最適化する［図2-6-1］**



Section

07 各章の進め方

ここまで、データサイエンスの必要性や定義に加えて、各章で学んでいく技術の概要を紹介してきました。データサイエンスのざっくりとした概要をつかめればよいなら、この第1章と第2章だけでも十分でしょう。

ただ、さらに踏み込んで「それぞれの技術をビジネス課題に対してどう適用すればよいか？」といった部分まで学んでいきたいという場合は、第3章以降は重要なパートです。

第3章以降はそれぞれ内容が独立しているので、もし前章を見て「この技術手法であれば自分の業務で活用できそうだ！」と思った部分があれば、その章を優先的に読み始めて大丈夫です。ただし、前章で学んだ内容をもとに進める部分も多少はあるので、特に優先度がなければ、第3章から順に読み進めることをおすすめします。

また各章の進め方は、次ページの［図2-7-1］のようなイメージで進んでいきます。本書の目的はデータサイエンスをビジネスで活用する基本知識とノウハウを得て、すぐに実務で活かせるようになることです。したがって、各章では具体的な業界のビジネスシーンを取り上げ、その課題解決のためにデータサイエンスの手法を学ぶ、という流れで進めていきます。

またより具体的なイメージが湧くように、各章の最後でExcelを用いたハンズオン演習をします。ハンズオン演習といっても、本書は各技術を用いて実際に何かを開発するデータサイエンティストやAIエンジニアを目指すものではないので、技術手法の実装自体はしません。取り上げたビジネスケースに関して、そのサンプルデータと、そのデータに対して実際に対象とする技術手法を適用した際の結果を用意しておきます。そのデータと結果を確認しつつ、実務に落とし込むためのアウトプットや、システムとの連携イメージをつかむことがゴールです。

技術手法を用いた結果を導くプロセスは本書では詳解していません。それには「Python」というデータサイエンスを実装するためのプログラミング言語が必要だからなの。さすがに Python による実装も学ぼうと思うと、もう 1 ～ 2 冊必要となるので、もし興味があれば、Python も学習してみて。

➲ 各章の進め方のイメージ［図2-7-1］

各章の進め方のイメージ	例）第4章の場合
①（あるビジネスケースを題材に）実務での活用シーンを紹介	・とある飲食店に関して、 ・課題解決のために「教師あり学習（回帰問題）」が必要
②そこで使うデータサイエンス技術手法を学ぶ	「教師あり学習（回帰問題）」の ・考え方や、基本的な手法 ・精度を評価するための評価指標
③ Excel を用いたハンズオン演習で実際の活用イメージを膨らませる	・POS データを使って ・教師あり学習の結果を解釈する

したがって、以降の章では、Excel があれば問題なくすべてを進められます。Excel は、Windows10 で Excel 2019 を用いた前提で解説していますが、Mac ユーザーでも問題なく使用可能です。本書では Excel を操作した内容がわかるように画面を掲載します。また Excel を操作する部分は全体の一部だけなので、もし Excel が手元になくとも、問題なく理解しながら進められるようになっています。

Chapter 3

基本的な可視化・統計手法を理解する

Section

01 店舗の売上実績を分析して現状を把握しよう

ここからは、具体的なビジネス課題に対するデータ活用を考えていきましょう。ちょうど、当社のクライアントである小売店舗のデータが手元にあるんだけど、どうやってデータを活用しよう？

せっかくいろいろな技術があることがわかったことだし、AI を使った高度な施策を提案したいですね！ 商品価格のダイナミックプライシングとか、需要予測による発注量最適化とかはどうですか？

たしかに、そういった高度な技術を活用した施策も面白そうね。でも、まずは地に足つけて、手元のデータをきちんと集計・可視化したり、統計的にどういった傾向があるかを観察したりして、現状を定量的に把握することから始めることが重要よ。

たしかに……

今後、機械学習などの高度な技術を活用していく際にも、まずはデータをしっかり理解できることが重要なの。データの集計・可視化、そして基礎的な統計処理は、データサイエンスのさまざまな技術の土台となるから、まずはそこからしっかり学びましょう！

ここで学ぶこと

- ☑ データの傾向をつかむ「集計（記述統計）」
- ☑ データを把握しやすくする「可視化」

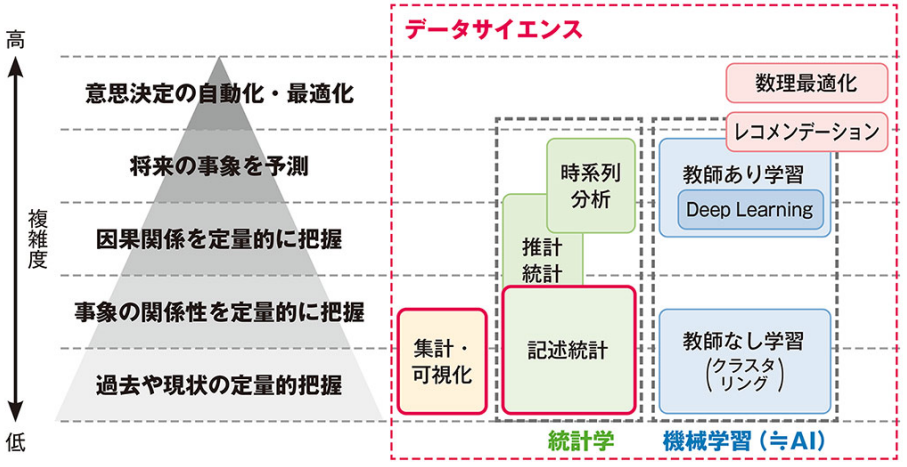
とある小売店舗の課題を考えてみよう

とある小売店 A 社のケースを考えてみましょう。この店舗では、これまで店長や店員の経験と勘で、店舗を運営してきました。しかし、新しいメンバーが運営業務に携わることなどを見据えて、誰にでも業務改善ができるような体制を構築したいと考えています。また経験と勘が当たらないケースも存在するはずです（もちろん、経験と勘がダメといっているわけではまったくなく、それらを体系化していくことに意義があるということです）。

その改善に向けた 1 つのアプローチとして、**店舗のデータを活用して、現状の把握や課題の仮説を抽出することができないか**と店長は考えました。これまでは、店舗データをしっかりと分析してこなかったのです。

なお、次の第 4 章以降では、機械学習などの高度技術を用いた施策適用を考えていくため、ビジネス課題をどうデータサイエンスで解けばよいかという問題設定を考えていきます。しかし今回は、まずは手元のデータをしっかりと把握し、どのような課題がありそうかという仮説出しに役立てる、という立ち位置で話を進めていくので、明確な KPI などは設けない形としましょう。もちろん、最終的にはコストや売上を改善することが目標となりますが、そのための第一歩としての現状把握をする、というイメージです。

➲ 本章で取り上げるトピックは主に集計・可視化と記述統計［図3-1-1］



※あくまでイメージです。実際は多種多様な手法や分野に分かれています

集計・可視化・基礎統計の重要性

近年ではAI、機械学習、ビッグデータという技術がバズワードとなって流行しています。そのため人間が特に何も考えなくとも、そのような技術がデータ分析をすべてこなしてくれるというイメージがありますが、そんなことはありません。実際本書でも次の第4章以降で昨今流行している機械学習などに関して、どのような技術なのか、そしてどのようにビジネス適用されているのか、ということを紐解いていきます。しかしすべてがシステマティックによしなにやってくれるということは（少なくとも現時点では）実現していません。

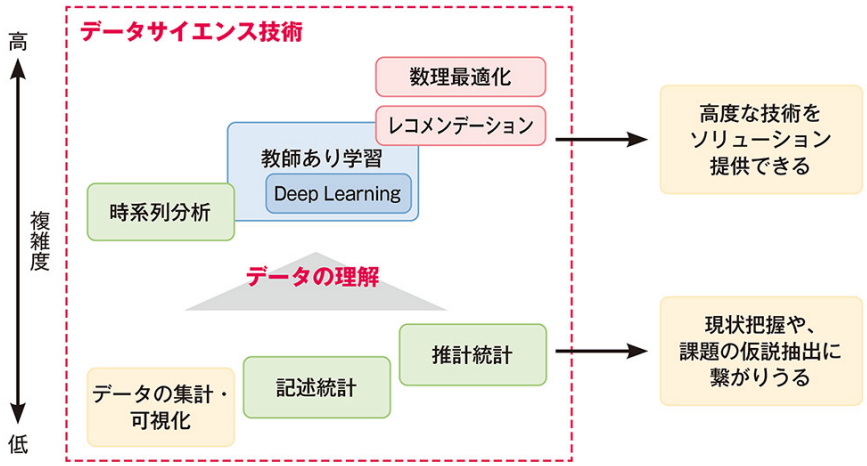
最前線に立つデータサイエンティストたちでさえ、データの中身をしっかり理解することに神経を注いでいます。そもそもAIと呼ばれている機械学習の技術をしっかり適用するためにも、まずはデータの中身を適切に理解しないといけません。

つまり、第4章以降で学ぶ機械学習のような技術を実装し、ビジネス適用するにあたっても、**まずは手元のデータを集計や可視化、基本的な統計処理を行って、そのデータを理解するというステップを外すべきではないでしょう。**

したがって、本章で学ぶデータの集計や可視化、基礎的な統計解析のようなトピックは、もしかしたらAIや機械学習に比べ有用性が劣ったり、技術として劣後する、などと思う人がいるかもしれませんが、そんなことはありません。どのような高度な技術を実装、適用していくにせよ、分析プロセス上の最初のステップとして非常に重要な手続きとなってきます。

わかりやすく理解するために、［図3-1-2］に、もう少し構造的にイメージを図示しています。データサイエンス技術は、前章でも説明したように非常に多岐に渡ります。しかし、“データ”サイエンスなので、基本的にはすべての技術に関して、使うデータのことを適切に理解したうえで高度技術も適切に開発・実装できると考えられます。

➲ データの理解がデータサイエンス技術の土台となる［図3-1-2］



※上記のマッピングはわかりやすさのために簡単化している部分があります。

一方で、データ理解のための集計、可視化、基礎的な統計解析は、いわゆるデータ分析（DataAnalytics）に相当するので、そこから得られた結果が示唆となり、現状の把握を通じた課題の仮説抽出につながりうると考えられます。仮説としての課題が浮かび上がってくれば、それをもとに新たな施策（打ち手）を検討できるでしょう[※1]。

もちろん高度な技術をソリューションとして開発し、業務やサービスに組み込みたい（例：レコメンデーションエンジンをECサイトに組み込みたい）というケースもあります。その場合は、直接的な技術はレコメンデーションエンジンなどの高度技術になりますが「そのエンジンを実装するための最初のステップとして、使用するデータを理解する」という流れになるでしょう。

それでは本章で、あらゆるデータサイエンス技術の土台となる集計、可視化、基礎的な統計解析部分について学んでいきましょう。特に統計解析部分は、それだけで本が数冊書けてしまうくらい幅広く奥深い分野なのですが、本書の紙面や目的に鑑みて、重要かつ基礎的な部分だけを抽出・要約して取り上げていきます。技術分野としては、データの可視化に加えて、記述統計を主にピックアップしたいと思います（推計統計にも非常に重要なトピックが満載なのですが、紙幅の関係上、本書では省略します）。

（※1）これらの分野に特化したデータ分析人材は「データアナリスト」と呼ばれます。

Section

02 要約統計量でデータの傾向をつかむ

それではさっそく、データ理解のためのデータ分析の中身に入っていきましょう。この第3章では、記述統計（集計）、データ可視化のトピックを1つのSectionごとに取り上げていきます。各Sectionにおいては、理論的な座学内容を学び、その後にExcelによる実践内容というステップで進めていきます。本章で取り上げる内容は、多くの部分に関してExcelで実装できるため、せっかくなのですべてExcelで実装することとします（次の第4章以降の機械学習などの技術は、Excelだけでは完結できないので、Pythonなどの言語により実装された結果を、Excelで解釈するという内容になります）。もちろん、本章のトピックも、より複雑なことをやろうとすると、プログラミング言語を利用したほうがよいケースがあります。

記述統計の必要性

さて、まずは「記述統計」と呼ばれる分野から学んでいきましょう。記述統計と聞くと難しそうですが、いわゆる「集計」とほぼ同義語と考えて差し支えないでしょう。これまでExcelに少しでも触れたことがあるなら、平均値を算出した経験があると思います。高校や大学で習ったという人も少なくないでしょう。

実は**平均を出すことも立派な統計手法の１つ**であり、記述統計に該当します。**記述統計は平均値を代表として、手元のデータからさまざまな値（それらを「要約統計量」といいます）を計算して、データの示す特徴や傾向を把握する手法**を指します。

「なぜ記述統計を使う必要があるのか？」というと、私は**データをくまなく全部チェックする手間を省くため**だと思っています。

もし時間が無制限であれば、すべてのデータを見てデータを把握できます。しかし、データが100行くらいならまだしも、1万行や10万行もあったら、

限られた時間内でデータを把握すること不可能です。そんなときに統計量を計算したり、データを可視化したりすると、データを１つ１つ見なくても傾向や特徴をつかめます。これが記述統計やデータ可視化を行うモチベーションであると考えてください。

とはいえ、データを直接見る必要がないかといわれると、そうではありません。むしろ生データを直接見ることは非常に重要です。実際の対象データが非常に大量であったとしても、そこからいくばくかをサンプリングしてみることはできるはずです。私自身、高度なアルゴリズムを開発する際も、一度生データを直接見ることにより、「なんか変な値が入っていそうだぞ？」など、集計だけではわからないようなことが肌感覚でわかり、その示唆をもってアルゴリズム開発の前のデータ前処理などに役立てられています。

もちろん大量のデータの中の一部しか見ていないので、バイアスがかかっている可能性もあります。したがって、生データを可能な限り観察しつつ、全体的な傾向を適切につかむための要約統計量も把握する、という形が望ましいでしょう。つまり、**真摯にデータと対峙するという姿勢が一番重要であると考えられます。**

さて、今回取り上げる要約統計量は、非常に重要かつ基本的な以下の指標となります。[※2]

➲ ここで扱う指標［図3-2-1］

- **平均値**
- **中央値**
- **分散、標準偏差**
- **最大値、最小値**

(※2) もちろんそのほかにもさまざまな統計量が存在します。

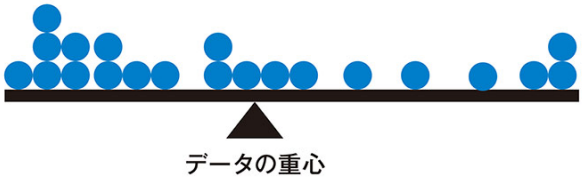
「平均値」を理解する

まずは、記述統計における統計量の代表例である「平均値」から見ていきましょう。平均値は知っている人が多いと思いますが、[図 3-2-2] のように定義されます。平均値はデータの「重心」であるともいえます（[図 3-2-3]）。

➲ 平均値の定義［図3-2-2］

$$\text{平均値} = \frac{\text{全データの合計値}}{\text{データ数}}$$

➲ 平均値はデータの重心である［図3-2-3］



平均値の注意点

一方で注意すべき点もあります。「1,2,3,4,5,6,7,1000」の平均値はぱっと見ていくつだと思いますか？

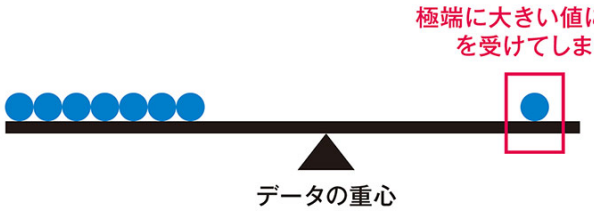
➲ 極端に大きい数がある場合の平均値［図3-2-4］

$$\text{平均値} = \frac{1+2+3+4+5+6+7+1000}{8} = 128.5$$

定義式にもとづいて計算すると、答えは128.5ですが、おそらく多くの方は、感覚的に3～5あたりを平均値にしたいと思うことでしょう。

このように、**平均値は極端に大きい値に影響されて、データ全体の中で相対的に大きめの値になってしまう**ことがあります。大きい値というのは、必ずしも正の方向だけではなく、負の方向に極端に大きな値（-10,000など）も含みます。

➲ 平均値は極端に大きい値に影響を受けやすい［図3-2-5］



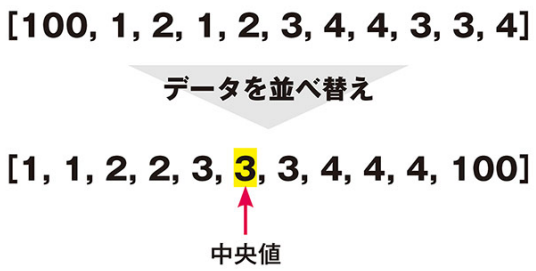
特にデータ数が少ないほど、極端に大きい値の影響を受けやすくなるため注意が必要です。とはいえデータの中身を全部確認してそのような問題がないことを確認してから平均値を算出するのは面倒でしょう。その場合、平均値だけを見るのではなく、後ほど紹介する「中央値」や「最小値・最大値」、「ヒストグラム」などと合わせて確認することが重要になります。

極端に大きい数字の影響を受けにくい「中央値」

平均値と並んで代表的な統計量である「中央値」について理解しましょう。

中央値は、データを小さい順（大きい順）に並べたときに、順位が中央である値のことを指します。次ページの［図3-2-6］のように対象となるデータを昇順で並べ替えて（ソートして）、データの数が5個であれば3番目のデータ、7個であれば4番目のデータが中央値となります。たとえば［図3-2-6］で、最も大きいデータである100が1,000や10,000に変わろうとも、データの中央の値は3のままです。このように、**中央値は極端に大きかったり小さかったりする数値の影響を受けにくい**というのが大きなメリットです。

➲ 中央値のイメージ［図3-2-6］



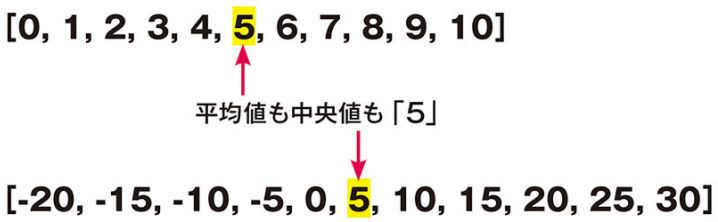
余談ですが、中央値はとてもわかりやすい指標なのですが、数式で表そうとすると結構難解になってしまいます。本書は数学の教科書ではないので式の定義は省略しています。

「分散」でデータのばらつきを定義する

平均値も中央値も、定義は異なりますが、要するに対象となるデータの「真ん中」の値を調べる統計量でした。しかし、真ん中の値を調べるだけで十分なのでしょうか？

［図 3-2-7］の 2 つのデータを見てください。どちらのデータも平均値、中央値が 5 となっていて、見かけ上は同じ傾向を示しているようです。しかしデータを見ると上のパターンに比べて下のパターンは、相対的にデータがばらついていると感じたのではないかと思います。これではデータの傾向を平均値、もしくは中央値だけで捉えることはできないでしょう。

➲ 平均値・中央値が同じでもデータのばらつきは異なる［図3-2-7］



［図 3-2-7］のような場合は、真ん中の値だけを見ていてもデータの傾向や全体像はつかめません。データの傾向をつかむには「データがどれくらいばらついているか？」という問いが重要となります。

そこで、データのばらつきを示すための統計量が必要となりますが、その1つが「分散」です。分散は、それによって得られた値を直接使うことはあまりありませんが、このあと紹介する「標準偏差」を計算するために必要な概念となります。**分散は、平均を中心にどのくらいデータがばらついているかを表す統計量**です。式の定義を説明すると、分散は各データと平均との差（偏差ともいいます）の2乗を合計してデータ数N-1（データの総数から1を引いた数）で割ったもの、となります。［図 3-2-8］に定義式を記載しておきましょう。

➲ 分散の公式［図3-2-8］

$$\text{分散} = \frac{\sum_{i}^{N} (i\text{個目のデータ} - \text{平均値})^2}{\text{データ数} N - 1}$$

分散を「標準偏差」に変換し、ばらつきを解釈する

分散の値の解釈が難しいのは、定義式をよく見てもらえるとわかると思いますが、**データ（正確にはデータ - 平均値）を「2乗」しているため**です。たとえば、売上個数の分散を計算すると、分散の値の単位は「個数」ではなく「個数の2乗」になってしまい、私たちが数値をうまく解釈できなくなってしまうのです。

これが先ほど触れた、分散で得られた値を直接的に使用する機会が少ない理由で、基本的には分散をもとにした標準偏差を使用します。標準偏差は、分散にルートをかけて単位を戻し、解釈できるようにします。

➲ 標準偏差の公式［図3-2-9］

標準偏差 ＝ $\sqrt{分散}$

分散はもとのデータの単位が2乗されているので、分散にルートをとれば、単位がもとに戻りますね。たとえば、面積はよく cm^2（平方センチメートル）で表されます。仮に $100cm^2$ の正方形があれば、一辺の長さはどうなるでしょうか？ 10cm × 10cm ですね。これを分散と標準偏差に置き換えると、分散が100であれば標準偏差は10ということになります。少し難しく数式で表せば、$\sqrt{100}=10$、ということになります。

もちろん、標準偏差も分散と同じように、**データのばらつきがどのくらい大きいか、小さいかを調べるために使用します。**

極端な値を探る「最大値」と「最小値」

最後に最大値と最小値も押さえておきましょう。文字通り、対象となる

・データの最も大きい値が「最大値」
・データの最も小さい値が「最小値」

となります。

最大値と最小値を確認することで、**「極端に大きく、もしくは、小さく外れている値がないか」を確認できます**。たとえば、以下のデータの集合を考えてみましょう（[図3-2-10]）。

➲ 最大値・最小値により極端に大きい・小さい値がわかる［図3-2-10］

[-100, 1, 2, 3, 4, 5,100]

→ 平均値 ＝ 2.1
→ 中央値 ＝ 3
→ 標準偏差 ＝ 57.8

ばらつきは大きそうだが、
外れ値の
イメージはつかない

平均値や中央値に加え、標準偏差を見る限り、たしかにデータはばらついていそうではあります。しかし、極端に大きい、小さい値がどうなっているかのイメージはつきません。このときに最大値・最小値を見ておけば、確実にそれらを検出できます。仮にこれらの値がおかしいと感じれば、もしかしたら間違って入力されてしまったデータかもしれない、と気づくことができます。

練習用ファイル：chap03_data_analytics / dataset.xlsx

実践 さまざまな要約統計量を求める

さて、ここまで紹介した統計量を、Excel を使って実践的に求めてみましょう。本章では、A 店の販売データを使用しましょう。本章のデータは chap03_data_analytics フォルダの dataset.xlsx ファイル[※3] です。

➲ 商品 ID ごとの販売データ［図3-2-11］

とある小売スーパー A店の、ある期間における販売データ

	A	B	C	D	E	F	G	H	I	J
1	商品ID	商品カテゴリ	価格タイプ	エネルギー(kcal)	商品価格(円)	売上個数(個)	売上金額(円)			
2	ID_001	その他	通常	83	144	66	9504			
3	ID_002	その他	通常	81	182	147	26754			
4	ID_003	その他	値引き	109	250	57	14250			
5	ID_004	その他	通常	82	170	42	7140			
6	ID_005	その他	通常	135	152	57	8664			
7	ID_006	その他	値引き	96	188	75	14100			
8	ID_007	その他	通常	94	155	81	12555			
9	ID_008	その他	値引き	110	321	153	49113			
10	ID_009	その他	値引き	128	179	99	17721			
11	ID_010	その他	通常	87	234	42	9828			
12	ID_011	その他	値引き	102	180	75	13500			
13	ID_012	その他	通常	104	130	102	13260			
14	ID_013	その他	通常	86	180	63	11340			
15	ID_014	その他	通常	94	156	78	12168			

（※ 3）Excel の解答例は「dataset_answer.xlsx」ファイルとして格納しています。

dataset シートに、今回使用する生データが格納されており、そのほかのシートは、今後の実践演習で用いていきます。

本データは、筆者がさまざまな小売業のオープンデータを参考にしながら、本章で学ぶ技術が網羅できるように独自で生成したダミーデータです。したがって、実際のビジネスやデータと傾向や特徴が乖離している部分も多少はあるかもしれませんが、ご了承ください。

本章では、この販売データを分析することで、商品の販売傾向を把握します。実際のビジネス背景がより詳細にわかっていないと、課題の詳細な仮説出しや施策案の検討は難しいはずなので、今回はあくまで「手元のデータからどういったことがいえそうか？」ということに主眼をおいていきましょう。

先ほど学んだ要約統計量は、Excel で求めることができます。本書は Excel でデータ分析の手続きを学ぶことには主眼をおいていない（あくまでデータサイエンスとはどういったものなのかを学ぶための本である）ので、以降で細かい Excel の操作方法には触れませんが、各要約統計量は、Excel に標準搭載されている、以下の関数で計算できます。

➲ **要約統計量を計算できる Excel の関数［図3-2-12］**

- **平均値：AVERAGE()**
- **中央値：MEDIAN()**
- **分散：VAR.S()**
- **標準偏差：STDEV.S()**
- **最大値：MAX()**
- **最小値：MIN()**

「基本統計量」シートを見てください。商品 ID ごとの売上個数のデータを dataset シートから転載しています。セル B2 ～ B683 に売上個数のデータが存在するので、それらの要約統計量を計算できます。実際に、要約統計量を計算してみましょう。試しに、最大値と最小値は空欄にしておいたので、関

数を用いて計算してみてください。

➲ 売上個数の要約統計量を計算する［図3-2-13］

	A	B	C	D	E	F	G	H
1	商品ID	売上個数(個)						
2	ID_001	66		Q. 売上個数の平均値・中央値・最大値・最小値を求めてみよう				
3	ID_002	147		平均値 =	83.20			
4	ID_003	57		中央値 =	81			
5	ID_004	42		分散 =	856.11			
6	ID_005	57		標準偏差 =	29.26			
7	ID_006	75		最大値 =	183			
8	ID_007	81		最小値 =	6			
9	ID_008	153						
10	ID_009	99						
11	ID_010	42						
12	ID_011	75						
13	ID_012	102						
14	ID_013	63						
15	ID_014	78						
16	ID_015	45						

［図3-2-12］の関数の引数を(B2:B683)にすることで求められる

この結果を見ると、下記のことがわかります。

・平均値と中央値はともに 81 ～ 83 付近にある
・ばらつきを表す標準偏差は約 30
・最小値は 6、最大値は 183 である

まず、平均値と中央値に大きな違いがない、つまり、**「極端に大きい／小さい値が含まれれば平均値は上がる／下がるが、中央値と比べて平均値はあまり変わらない。ということは、売上個数に極端に大きな値（外れ値）はない」**であろうと推察できます。

そして、標準偏差が約 30 で売上個数の平均が約 83.2 であるため「多くのデータが 83.2 ± 30（つまり 53.2 ～ 113.2）に存在する」と解釈します。 注意すべき点としては、「すべてのデータが約 83.2 ± 30 に存在しているわけではない」ということです。

標準偏差は、「全データがどの程度平均値からばらついている傾向がある

か」を示している指標です。そのため、平均値83.2から50以上離れているデータもあれば、逆にほとんど離れていないデータもあるということです。あくまで平均的に、83.2 ± 30に散布している傾向にある、という捉え方、解釈をしてください。

また最小値や最大値を見ても、外れ値などのおかしい値はなさそうです。売上個数は多くが83.2 ± 30に存在するが、最小で6個、最大で183個売れているのか、と考えられます。仮に、最小値がマイナスになっていたら、データの定義上明らかにおかしい異常値であるとわかります。また最大値は理論上どのような値でも許容されますが、このデータの傾向で、最大値が10,000になっていたらおかしいと考えられるでしょう。

このように、要約統計量だけでも、データをある程度深く観察することができます。なお、Excelには「分析ツール」という機能があり、それを使用すれば、一発でこれらの要約統計量を算出できます。本書では取り扱いませんが、興味のある方はぜひ調べてみてください。

統計的観点で、少し発展的な話をすると、今回のように「平均値＝83」と、ある1点を推定する方法は「点推定」と呼ばれます。そのような“点”推定ではなく、「区間推定」という方法論も存在します。区間推定は文字通り“区間”を推定するため、「（統計的に鑑みて）平均値はxxからyyの範囲に収まる可能性が高い」といったような推定ができるようになります。区間で推定することができ、下限や上限の情報も含むことができるので、リスクマネジメントなどの観点から使われることもあります。

推定の方法は統計的な知識が必要なので割愛しますが、そのような方法論が存在するということは頭の片隅に入れておくとよいでしょう。

ここで学ぶ内容は、姉妹本である『統計学の基礎から学ぶExcelデータ分析の全知識』でより詳しく取り上げています。姉妹本を読了している方は、おさらいのつもりで読み進めてください。

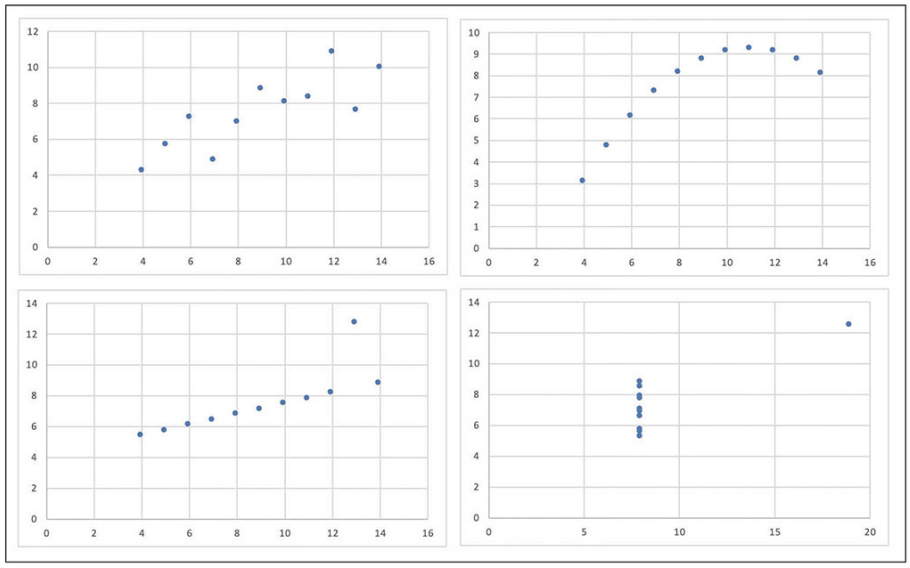
Section

03 実務で使えるデータ可視化

なぜデータの可視化が必要なのか

前 Section では平均値や標準偏差といった重要な要約統計量を紹介しました。しかし、「データの傾向をつかむ」という観点からは、統計量のみならずデータの可視化も理解しておくべきです。データの可視化とは、データを視覚的に把握できるようにグラフなどにすることです。その一例ですが、[図3-3-1] を見てください。この図は「散布図」と呼ばれる可視化方法です。詳しくは後ほど紹介しますが、横軸 x と縦軸 y の 2 次元空間にデータを並べています。これらの散布図を見て、おそらく全然違う傾向を持つデータだと感じたのではないでしょうか？

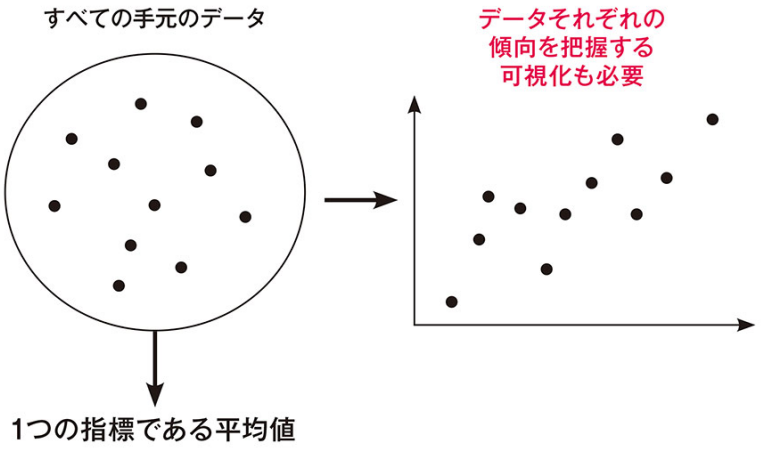
➲ アンスコムの例によるデータの可視化［図3-3-1］



実は、**すべての図で「x 軸の平均は 9、標準偏差は 3.32」「y 軸の平均は 7.5、標準偏差は 2.03」なのです！**これは「アンスコムの例」と呼ばれ、フランク・アンスコムという学者によって紹介された数値例から引用しているのですが、データの統計量だけではなく、しっかりデータを可視化して傾向をつかむことの重要性を強く示唆しています。

要約統計量とデータ可視化はセットといってもいいかもしれません。改めて「なぜデータを可視化するのか？」というと、**データの傾向をより正確に把握するため**です。「平均値」などの要約統計量は指標の 1 つであり、データそれぞれの傾向を表す「可視化」も必要なのです。

➲ データの把握には可視化も必要［図 3-3-2］



本 Section では、特に知っておいてほしい［図 3-3-3］に挙げた 5 つの可視化の手法を紹介します。どれも実務で使用頻度の高いものです。私や私が所属してきたデータサイエンスチームが、実際の案件やプロジェクトにてよく使用していた可視化手法を中心に取り上げているので、皆さんも「実際に自分たちの現場で使えないか？」といった視点で見てください※4。

（※ 4）もちろんほかにもよく使われるさまざまな可視化手法がありますが、紙幅の関係上、優先度を付けて絞りました。興味があればいろいろと調べてみてください。

➲ 5 つの可視化手法［図3-3-3］

・ヒストグラム
・棒グラフ
・ヒートマップ
・散布図
・相関行列（相関係数）

データの分布の形状を把握する「ヒストグラム」

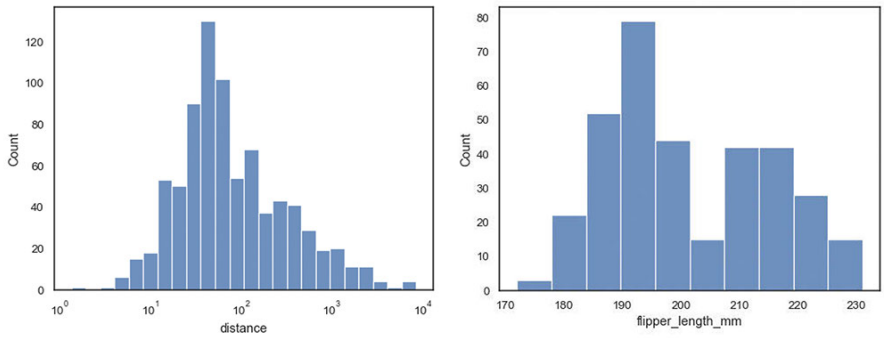
可視化の手法を解説する前提として、可視化の対象となる値（「変数」）は、基本的に以下の 2 種類に分類されることを知っておきましょう。

➲ 変数には 2 種類ある［図3-3-4］

1. **連続変数：**
   値が連続的に変化する変数（売上、個数など）
2. **カテゴリカル変数（離散変数ともいう）：**
   値と値の間に距離がない変数（名前、性別、商品分類など））

なぜここで種類を紹介したかというと、変数の種類によって、可視化の手法を考えていくためです。具体例としてヒストグラムから見ていきましょう。「ヒストグラム」は非常によく使われる代表的な手法です。**1 つの連続変数の分布（データの傾向）を見たいとき**に用いられ、データがある値の範囲内にいくつ存在しているかを把握できます。たとえば、商品単価や販売個数といった 1 つの連続変数に関して、それがどのように分布しているのかを把握することができます。

➲ ヒストグラムのイメージ［図3-3-5］



1つの連続変数の分布（データの傾向）を見たいときに用いる

参照：https://seaborn.pydata.org/index.html ※5

もう少し詳細に定義を見ると、ヒストグラムの各棒（**「ビン」**といいます）は、ある値からある値までの範囲を示しており、対象の変数に関して、各ビンにいくつのデータが存在しているかが示されています（場合によってはカウントではなく割合のケースもあります）。

このビンの幅、もしくはビンの数は変更できます。ビンの幅を小さくする＝ビンの数を増やすことにより、より詳細にデータの傾向を把握できるでしょう。この調整は対象の変数を実際に可視化しながら行うことになります。

ヒストグラムを見る際の論点を整理しておきます。基本的には以下の点を意識しながら、ヒストグラムを眺めてみましょう。

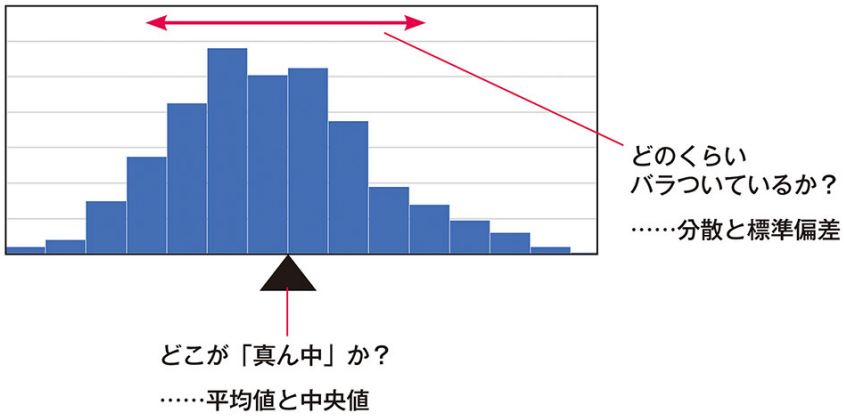
➲ ヒストグラムの見方［図3-3-6］

- **山がいくつあるか？**
- **外れ値がないか？**
- **データの「中心」はどのあたりか？**
- **データの「ばらつき」はどの程度か？**

（※5）以降の各種可視化手法のイメージの紹介時はすべて記載の参照元の図を引用しています。

特に最後の2つに関しては、前 Section で学んだ「平均値」「中央値」「分散」「標準偏差」と関係してきます。つまり、**ヒストグラムの中心が平均値や中央値に対応し、ヒストグラムのばらつきが分散や標準偏差と関係している**のです。この関係性のイメージを頭の中に入れておきましょう。

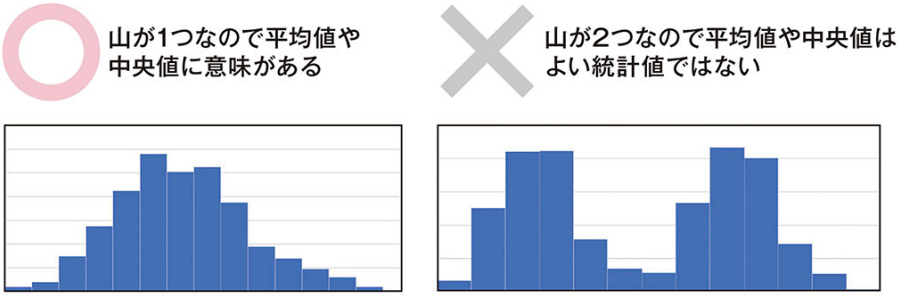
➲ ヒストグラムと基本統計量の関係［図3-3-7］



ヒストグラムの山の数も、基本統計量との関係があります。たとえば次ページの［図 3-3-8］の右側のように山が2つある場合は、平均値や中央値を計算してしまうと、それらの値は山と山の間あたりの値になってしまうため、データの傾向を適切に表していない可能性が高くなります。そのような、山が1つのきれいな（釣り鐘状の）分布となっていない場合に「必ずこう対処すべき」というやり方があるわけではありません。基本的にはデータを細かく見ることでどういう状態になっているかを確認しましょう。

たとえば、山が2つあるので、異なる集団が含まれてる可能性があります。男性と女性、年齢が低い／高い層で異なる性質を持っているために山が分かれている、といった具合です。したがって、ヒストグラム上で山が分かれている部分の値を閾値としてデータを区切ったときに、年齢や性別などのさまざまなデータ列（変数）で集計して、どのような変数が分布の形状に影響を与えているのか、などを調べるようにしましょう。そのような変数が見つかれば、その変数にもとづいてデータの集団を2つに分けてそれぞれで分析を行う、などの対応が考えられます。

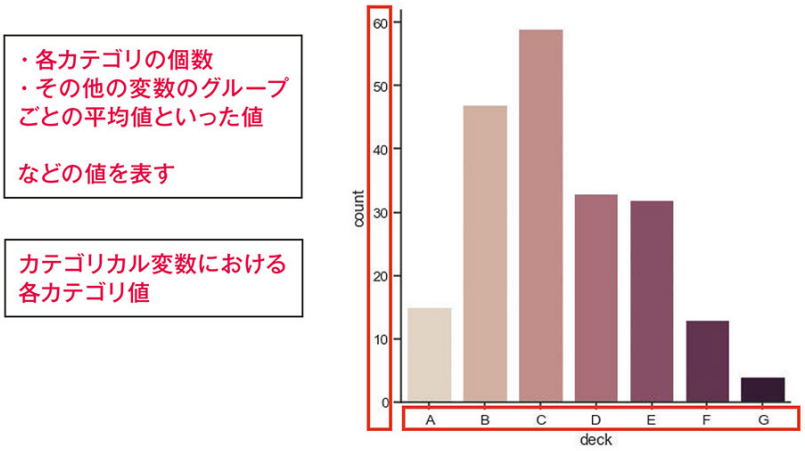
➲ ヒストグラムの山の形状を確認［図3-3-8］



カテゴリ間の値を比較する「棒グラフ」

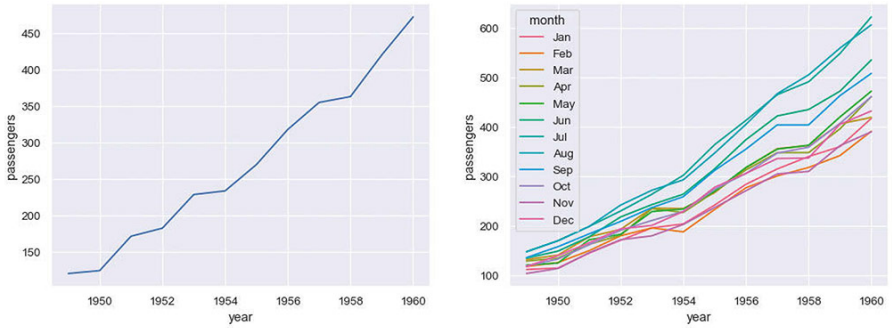
続いては「棒グラフ」です。比較的馴染みのあるグラフだと思いますが、ここでその機能を確認しておきましょう。棒グラフは基本的に、**カテゴリカル変数におけるカテゴリ間の、値の大小を比較したい**ときに使用します。［図3-3-9］のイメージのように、横軸はカテゴリカル変数の値（グループ、水準）であり、縦軸はデータの個数を表す場合とデータの値を表す場合があります。つまり、「カテゴリカル変数×連続変数」を可視化する際に使います。また特に「（［図3-3-9］における）縦軸が何を表しているか」は必ず注意して見ておくようにしましょう。

➲ 棒グラフのイメージ［図3-3-9］



しばしば「棒グラフ」と「折れ線グラフ」の使い分けに困るかもしれませんが、折れ線グラフは基本的に（時間などを通じた）推移や変化を表したいときに使います。［図 3-3-10］のようなイメージで、たとえば株価の推移などといったものがわかりやすいでしょう。

➲ 折れ線グラフは時間による推移を示すときに用いる［図3-3-10］



横軸に時間などの変化を表す連続数値を取り、縦軸に一連の関係性のある連続数値をとります。

つまり、横軸に定義される変数の値が独立した集団のデータであれば「棒グラフ」、一連の関係性がある集団のデータであれば「折れ線グラフ」が向いている、と考えるのがおすすめです。

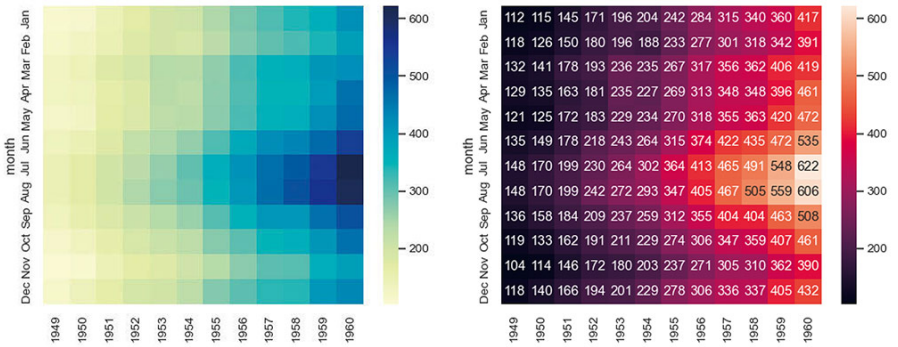
したがって、商品 A、B、C の売上個数のような場合は、それぞれが独立しているため、棒グラフが向いています。また、7 日前、6 日前、5 日前……1 日前の株価のような場合は、一連の関係性がある集団のデータであるため、折れ線グラフが向いているといえるでしょう。したがって棒グラフを使用したいときは、推移や変化を表すわけではなく、カテゴリカル変数におけるカテゴリ間での比較ということをきちんと表しているかどうか、を確認しておきましょう。

行列型でデータの特徴を把握できる「ヒートマップ」

ヒートマップは、2 次元データを色の濃淡で表したものです。ヒートマップも考え方はそこまで難しくありません。行列の形において、2 つのカテゴリカル変数を行と列として、ある行ある列ごとに、1 つの連続変数の値を入れ込みます。

しかしそれだけでは可視化したことにならないので、数字の大小で色の濃淡をつけて、どの部分の値が大きいか、または小さいかを視覚的に把握できるようにするのです。

➲ ヒートマップのイメージ［図3-3-11］



たとえば、横軸に年、縦軸に月、行列の値に年ごと月ごとの平均販売個数、といった具合でプロットすることにより、「年・月・販売個数」という **3 次元のデータを 2 次元空間上で表すことができます**。

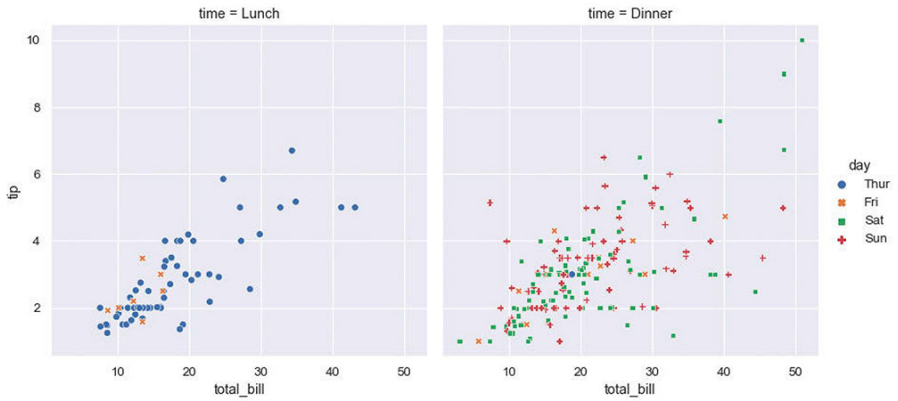
またヒートマップの色を工夫することで、見栄えもよくなり、解釈もしやすくなります。たとえば［図 3-3-11］のように単一色のグラデーションにすることで、どの部分が値が大きいのか、小さいのか、という傾向がわかりやすくなりますね。その際の色は何でもよいですが、赤や青などのメインカラーや、あるいは会社のロゴ色やチームで決めている標準色などがあれば、それらを使用するのがよいでしょう。

2つの連続変数の傾向を把握する「散布図」

たとえば、売り場における商品の占有率と、その商品の売上数のような「連続変数×連続変数」の関係性を知りたいときはどうすればよいでしょうか？そのときは「散布図」を使用しましょう。作り方はとても簡単です。対象とする２つの連続変数を横軸と縦軸において、それぞれのデータを点にして値に応じて並べるだけです。

非常にシンプルですが、これによって両変数がどのくらい「相関」しているかが可視化できます。**相関とは、ある変数の値が増加または減少すれば、もう他方の変数の値もそれに伴って増加、減少する関係性のこと**です。その関係性がどのくらいの強さであるかが、相関の強さを示すこととなります。統計学においては非常に重要な指標の１つです。実際に相関の強さを図るための指標を「相関係数」といい、次節で紹介します。

➲ 散布図のイメージ［図3-3-12］



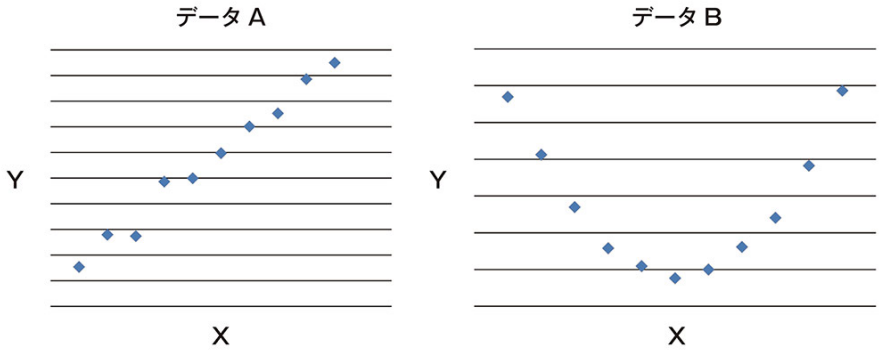
散布図が右肩上がりの傾向になっている場合は、片方の変数が高いと他方の変数も高くなるような「正の相関関係」にあることがわかります。一方で右肩下がりであればその逆で、「負の相関関係」であるといえるでしょう。

変数間の相関を示す「相関係数」

先ほどの「相関」についてもう少し深堀りしてみましょう。その延長線上で「相関行列」を紹介しますが、そのためにはまず「相関係数」を知っておく必要があります。結論からいうと、**統計学の世界で「相関」といった場合、通常はデータが直線状に並んでいることを表します**。

だとすると、［図 3-3-13］の散布図 A、B に関して、統計学的に「相関」関係があるのは、どちらのデータだと思いますか？

➲ データ A と B の散布図［図3-3-13］

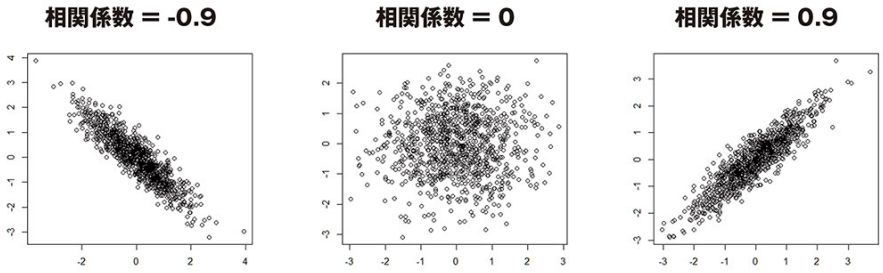


答えは、直線状にデータが並んでいる A となります。なぜなら、繰り返しますが相関関係というのはデータが「直線状」に並んでいることを指すためです。

そして、その相関関係の強さを表すための指標として、「相関係数」というものがあります。次ページの［図 3-3-14］に示したように**相関係数は -1 から +1 までの間の値**として表され、x 軸の値が大きいデータほど y 軸の値も大きい右肩上がりの散布図になっている場合は、相関係数が +1 に近づきます。その逆で、x 軸の値が上がると y 軸の値は下がってしまう右肩下がりの散布図の場合は、相関係数が -1 に近づきます。そして、x 軸と y 軸にまったく右肩上がり、右肩下がりのような関係性がない場合、相関係数は 0 に近づいていきます。

したがって、**相関係数が +1 または -1 に近づいていくほど「相関が高い」**といえます。逆に**相関係数が 0 に近いと「相関が低い」**ということです。

➲ 相関関係のイメージ［図3-3-14］



相関度合いを表す「相関係数」は-1から+1で表される

なお、-1 と +1 の場合はデータが完全に直線上に重なった状態となるため、このグラフではイメージが伝わるように -0.9 と 0.9 にしてある

相関係数がどのように計算されているかに関しては、今回は紙面や本書の主旨の関係から省略します。コンセプトとしては、x 軸の変数と y 軸の変数に関して、［図 3-3-15］のような、両変数間の分散（「共分散」）を計算することで、相関係数を定義しています。この相関係数も、（平均値や標準偏差などに加え）記述統計における重要な指標の 1 つだと考えてよいでしょう。

➲ 変数間の分散を計算して相関係数を定義する［図3-3-15］

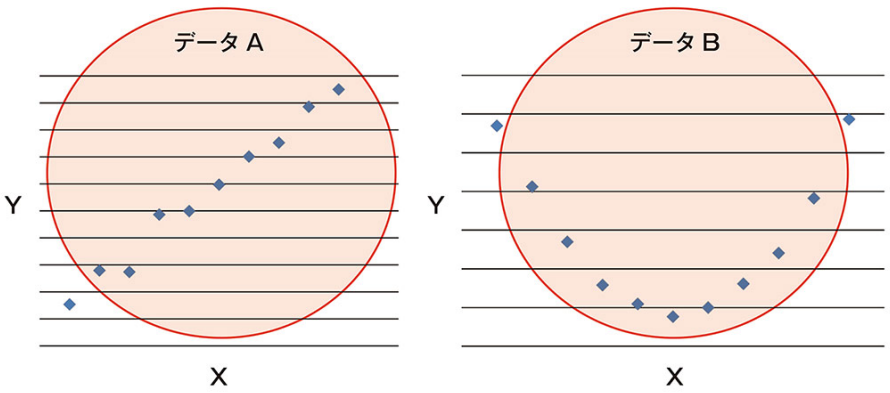
・右肩上がり＝正の関係にあるほど大きく、
・右肩下がり＝負の関係にあるほど小さくなる

ただしこの相関係数には注意点があります。改めて前ページの［図 3-3-13］のデータ A と B で、X と Y に「関係」があるのはどちらだと思いますか？ たしかにデータ B の相関係数は低くなりますが、では X と Y に関係がないかといわれると、そんなことはないケースもあります。たとえば、気温と電気の使用量はどのような関係でしょうか？ 真冬の寒いときや真夏の暑いときはどちらもエアコンをつけますよね？ すると電気使用量は上がりそうです。一方で春や秋の比較的気温が穏やかなときの電気使用量は少なくなりますね。まさにデータ B のような傾向を表すでしょう。この場合、気温と電

気使用量は「相関」はないかもしれないが、「関係」はあるといえそうです。

したがって、相関係数は１つの値として算出できて便利なのですが、統計学における相関係数が低いからといって、２つのデータの間に関係がないとはいえないのです。なので、相関係数という便利な指標を利用しつつ、新たなデータを観察する際は、必ず散布図も描いて確認するのがいいでしょう。

➲ 相関係数が低くとも関係性がないとはいえない［図3-3-16］



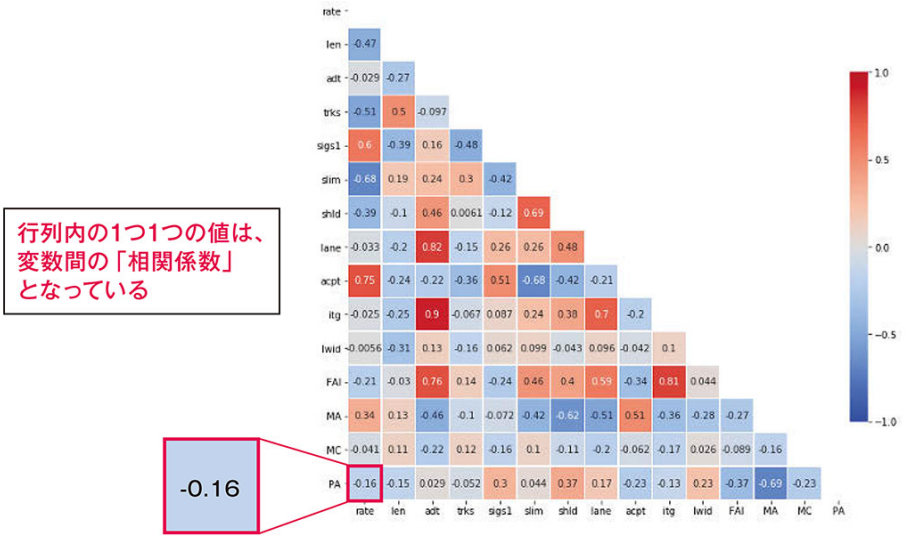
正確には、線形ではなくとも（たとえば［図 3-3-16］のデータＢのような二次関数の形状となっているデータでも）関係を捉えられるような発展的な相関係数は、統計学の学問においては存在します。しかし今回の主旨とは外れるので、詳細には取り上げません。

変数間での相関係数が一目瞭然「相関行列」

さて相関の延長上にあるのが「相関行列」です。相関係数がわかってしまえば簡単です。相関行列とは、対象とするすべての連続変数に対して、それら変数間のすべての組み合わせにて計算される相関係数を行列の形式で表したものです。

さらに、行列内に値が格納されているだけだと少し見にくいので、先ほど紹介したヒートマップを駆使することで、わかりやすく可視化することが多いです。

➔ 相関行列のイメージ［図3-3-17］



ヒートマップでは、分析したい対象すべての変数を行と列に配置し、それぞれの組み合わせに対する相関係数を表示します。一発ですべての組み合わせの相関係数がわかり、とても便利な可視化方法といえるでしょう。

対角線上の同じ変数同士は、当然同じ変数なので相関係数は1となっています。それ以外の相関係数に関して、特に相関係数が1や-1に近い、つまり正もしくは負の相関を持っている変数同士を見つけることで、何か新しい発見を見出すことができるかもしれません。

なお、ヒートマップの色に関しては、基本的に自由に設定して問題ありません。ただ相関係数は正か負の値かで解釈がかなり異なってくるので、0から1に近づくほど赤くなり、0から-1に近づくほど青くなる、といったようなカラースケールが解釈しやすいでしょう。

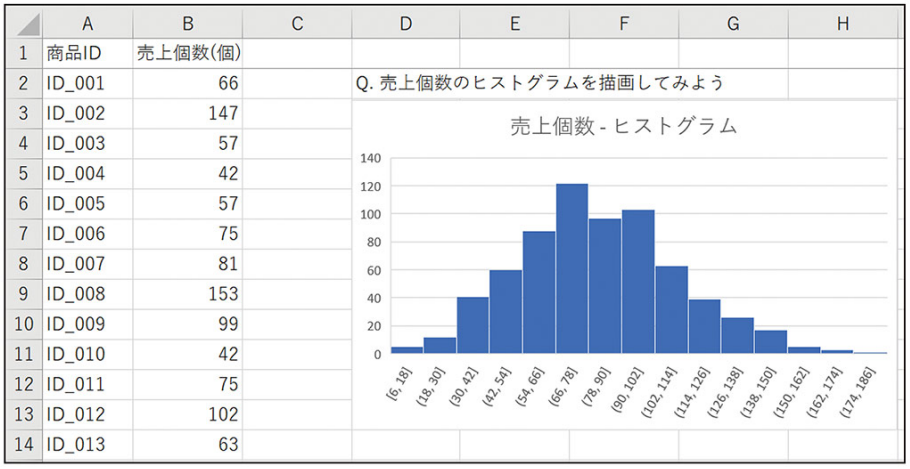
練習用ファイル：chap03_data_analytics/dataset.xlsx

実践 さまざまな可視化を試してみる

さて、ここまで学んだ可視化手法を、Excel で簡単に描画してみましょう。dataset.xlsx に、各可視化手法のシートを合計 5 枚用意しています。

まずは「ヒストグラム」シートから見ていきましょう。売上個数のヒストグラムを描くために、セル B2 ～ B683 を選択し、［挿入］タブから［統計グラフの挿入］→［ヒストグラム］を選択すると、簡単にヒストグラムを描くことができます。

➲ 売上個数のヒストグラム［図3-3-18］



	A	B	C
1	商品ID	売上個数(個)	
2	ID_001	66	
3	ID_002	147	
4	ID_003	57	
5	ID_004	42	
6	ID_005	57	
7	ID_006	75	
8	ID_007	81	
9	ID_008	153	
10	ID_009	99	
11	ID_010	42	
12	ID_011	75	
13	ID_012	102	
14	ID_013	63	

ヒストグラムを見ると、売上個数は［78, 90］を中心に、概ね左右対称に分布しているのがわかります。ちなみに統計学では、**左右対称で釣鐘状となる分布のことを「正規分布」といいます。**このヒストグラムは、正規分布に近い形であるといえます。

また［図 3-3-18］のヒストグラムの左側を見ると［6, 18］となっており、6 以下の数値は存在しないこともわかります。売上個数は正の値しか取り得ないため、「今回の売上個数には異常な値はなさそうである」と考えられます。なお、**もし売上個数のヒストグラムでマイナスの範囲に値が存在していたら、異常な値が入っているといえます。その原因を探らなければなりません。**

またヒストグラム全体を俯瞰すると、平均的には約 60 ～ 100 個あたりの売上個数となっている商品で構成されており、売上個数が非常に多い超人気商品はほぼない傾向がわかります。つまり、もちろん多少のばらつきはありますが、ある一部の商品に依存しておらず、販売商品がバランスよく販売されていそうであることが読み取れます。

続いては棒グラフです。「棒グラフ」シートを開いてください。今回は、商品カテゴリごとの、売上個数の平均値を棒グラフで描画してみましょう。セル E3 ～セル F14 を選択し、[挿入]タブの[縦棒 / 横棒グラフの挿入]→[2-D 縦棒]の［集合縦棒］をクリックします。これで商品カテゴリごとの平均売上個数の棒グラフが作成できました。

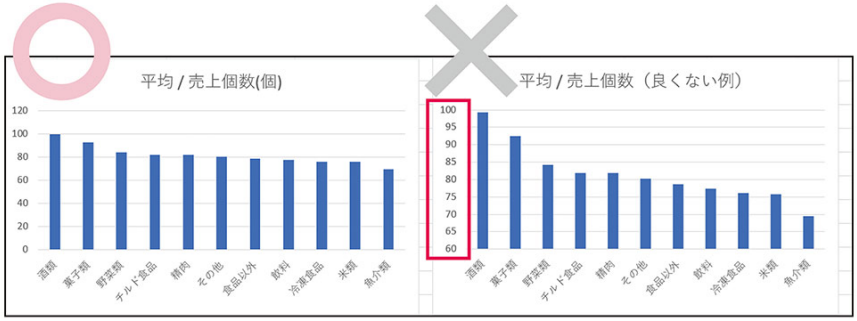
➲ 商品カテゴリごとの平均売上個数の棒グラフ［図3-3-19］

	A	B	C	D	E	F	G	H	I	J	K	L
1	商品ID	商品カテゴリ	売上個数(個)									
2	ID_001	その他	66		タイプ毎の売上個数の平均値			Q. 売上個数のタイプ毎の平均値の棒グラフを作成しよう				
3	ID_002	その他	147		行ラベル	平均 / 売上個数(個)						
4	ID_003	その他	57		酒類	99.39130435						
5	ID_004	その他	42		菓子類	92.48888889						
6	ID_005	その他	57		野菜類	84.22058824						
7	ID_006	その他	75		チルド食品	81.83333333						
8	ID_007	その他	81		精肉	81.80357143						
9	ID_008	その他	153		その他	80.28571429						
10	ID_009	その他	99		食品以外	78.71641791						
11	ID_010	その他	42		飲料	77.33333333						
12	ID_011	その他	75		冷凍食品	76.07865169						
13	ID_012	その他	102		米類	75.77419355						
14	ID_013	その他	63		魚介類	69.42857143						
15	ID_014	その他	78		(空白)							
16	ID_015	その他	45		総計	83.20381232						
17	ID_016	チルド食品	54									
18	ID_017	食品以外	78									

棒グラフを見ると、酒類や菓子類の平均売上個数が比較的高いことから、お酒やお菓子を買う需要が高いのかもしれません。これらの商品の拡充に注力するといった打ち手も考えられるでしょう。

さて、少し棒グラフに関して注意点です。次ページの［図 3-3-20］の 2 つの棒グラフを見てください。両者で見た目が違いますが、実は同じ情報を示しています。では何が違うかというと、縦軸（y 軸）の原点が左図は 0 で、右図は 60 であるという点です。

➲ **棒グラフの原点に注意［図3-3-20］**



右図のように縦軸を途中の値からとすれば微細な差でも重大な差であるかのような印象を与えられます。したがって、作成する際も図を見る際も、特に縦軸のスタート値には注意が必要です。原点は 0 にするのが原則です。

続いて「ヒートマップ」シートを見ましょう。ここでは、商品カテゴリごと・価格タイプごとの、平均売上個数のヒートマップを描画します。セル G22 ～セル H32 には、あらかじめタイプごと、価格表示ごとの平均売上個数の表が作ってあります（価格表示は「値引き」と「通常」が表示）。この表をヒートマップにしてみましょう。セル G22 ～セル H32 を選択し、［ホーム］タブの［条件付き書式］にある［カラースケール］から好きな色を選択してみましょう。対象とした部分に関して、値が大きいセルがより赤く濃く色づけされているのがわかるでしょうか？（[図 3-3-21]）

これで、だいぶわかりやすく結果を見ることができますね。列で見ると、全体的に通常価格の商品より値引きした商品のほうが売上個数は多そうです。また詳細にみると、酒類と菓子類などが高く、菓子類の値引きの売上個数が 101 と、一番高いセグメントであるといえそうです。

一方で通常価格時の米類や魚介類商品は少々足を引っ張っていそうでしょうか。これらのセグメントは、より一層の強化（あるいはこれらの強化を見捨て、酒類や菓子類へ注力するなどの選択と集中）が必要かもしれません。

➲ タイプごと価格表示ごとのヒートマップ［図3-3-21］

	A	B	C	D
1	商品ID	商品カテゴリ	価格タイプ	売上個数(個)
2	ID_001	その他	通常	66
3	ID_002	その他	通常	147
4	ID_003	その他	値引き	57
5	ID_004	その他	通常	42
6	ID_005	その他	通常	57
7	ID_006	その他	値引き	75
8	ID_007	その他	通常	81
9	ID_008	その他	値引き	153
10	ID_009	その他	値引き	99
11	ID_010	その他	通常	42
12	ID_011	その他	値引き	75
13	ID_012	その他	通常	102
14	ID_013	その他	通常	63
15	ID_014	その他	通常	78
16	ID_015	その他	通常	45
17	ID_016	チルド食品	通常	54
18	ID_017	食品以外	通常	78
19	ID_018	食品以外	通常	69
20	ID_019	冷凍食品	通常	63
21	ID_020	野菜類	通常	108
22	ID_021	野菜類	通常	72
23	ID_022	菓子類	値引き	105
24	ID_023	チルド食品	通常	54
25	ID_024	食品以外	通常	75
26	ID_025	冷凍食品	通常	93
27	ID_026	その他	値引き	99
28	ID_027	食品以外	通常	45
29	ID_028	冷凍食品	通常	72
30	ID_029	精肉	通常	78
31	ID_030	野菜類	通常	75
32	ID_031	野菜類	通常	90
33	ID_032	菓子類	通常	111

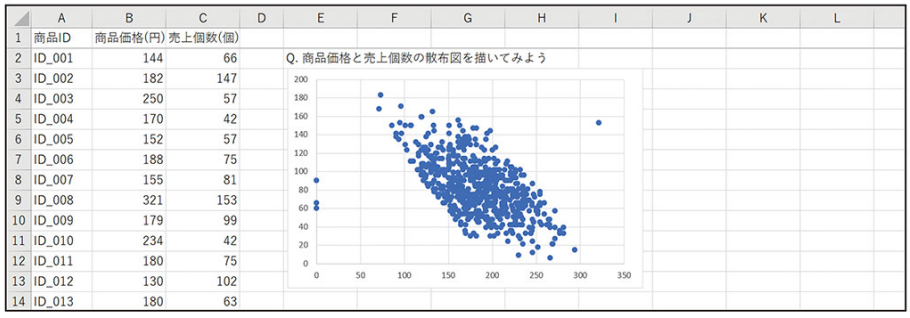
平均 / 売上個数(個)	列ラベル			
行ラベル	値引き	通常	(空白)	総計
酒類	99.000	99.529		99.391
菓子類	101.667	89.152		92.489
野菜類	90.237	81.888		84.221
チルド食品	91.909	77.400		81.833
精肉	89.647	78.385		81.804
その他	93.000	75.200		80.286
食品以外	97.588	72.300		78.716
飲料	87.857	72.581		77.333
冷凍食品	74.455	76.612		76.079
米類	94.500	69.261		75.774
魚介類	93.000	51.750		69.429
(空白)				
総計	91.7936508	79.9107505		83.203812

Q. タイプ毎・価格表示毎の売上個数のヒートマップを作成してみよう

	値引き	通常
酒類	99.000	99.529
菓子類	101.667	89.152
野菜類	90.237	81.888
チルド食品	91.909	77.400
精肉	89.647	78.385
その他	93.000	75.200
食品以外	97.588	72.300
飲料	87.857	72.581
冷凍食品	74.455	76.612
米類	94.500	69.261
魚介類	93.000	51.750

4つ目は散布図です。「散布図」シートでは、商品価格と売上個数の関係性を可視化してみましょう。B列とC列を選択し、［挿入］タブの［散布図(X, Y) またはバブルチャートの挿入］から［散布図］を選択します。横軸にB列の商品価格、縦軸にC列の売上個数として、データがすべてプロットされているのがわかるでしょう。

➲ 商品価格と売上個数の散布図［図3-3-22］



	A	B	C
1	商品ID	商品価格(円)	売上個数(個)
2	ID_001	144	66
3	ID_002	182	147
4	ID_003	250	57
5	ID_004	170	42
6	ID_005	152	57
7	ID_006	188	75
8	ID_007	155	81
9	ID_008	321	153
10	ID_009	179	99
11	ID_010	234	42
12	ID_011	180	75
13	ID_012	130	102
14	ID_013	180	63

Q. 商品価格と売上個数の散布図を描いてみよう

この結果を見るとどうでしょう。まず全体を見ると、点の塊が右下に傾いていることがわかります。このことから、商品価格が上がるほど、売上個数が下がっていく傾向が見てとれます。一般常識的には、商品価格（単価）が高いほど、その分売れる個数は少なくなりそうです。そう考えると、今回のデータも概ね予想していた通りの傾向になっていそうです。 しかしよく見ると、商品価格が0円のデータが存在します。0円ということは基本的にはありえないので、もしかしたら異常値が潜んでいるのではないか、と疑うことができます。また一方で、商品価格も高く売上個数も高い、点の集団から外れているようなデータも見受けられます。これは全体の傾向からかけ離れた商品の可能性があり、より深く掘り下げてみる価値がありそうです。このように、可視化によってデータを注意深く観察することにより、新たな発見が生まれますね。

最後に相関係数、相関行列を見ておきましょう。「相関行列」シートを開きましょう。まずは先ほど見た商品価格と売上個数の相関係数を計算してみます。散布図では右肩下がりの傾向だったので、相関係数も負の値になりそうでしょうか。

相関係数を求めるには、CORREL関数を使います。引数には相関関係を求めたいデータが入力された範囲を指定します。ここではC列（セルC2～セルC683）とD列（セルD2～セルD683）が対象なので、セルH3に「=CORREL(C2:C683,D2:D683)」と入力して［Enter］キーを押します。

すると、約－0.54と出力されたでしょうか。つまり「商品価格と売上個数の相関係数は－0.54である」といえます。

➲ **商品価格と売上個数の相関係数［図3-3-23］**

	A	B	C	D	E	F	G	H	I	J
1	商品ID	エネルギー(kcal)	商品価格(円)	売上個数(個)	売上金額(円)					
2	ID_001	83	144	66	9504		Q. 商品価格と売上個数の相関係数を求めてみよう			
3	ID_002	81	182	147	26754		相関係数 =	-0.5426		
4	ID_003	109	250	57	14250					
5	ID_004	82	170	42	7140					

今回は連続変数が４つあるので、それぞれの組み合わせに関して相関係数を計算することができます。その計算結果は［図 3-3-24］にあります。ただし、より変数が多くなるとすべて計算するのが大変になるので、実務的には［データ分析］ツールを使うことが多いです。

もし Excel で［データ分析］ツールを設定できている場合は次のようにします。［データ］タブの［データ分析］をクリックし、ダイアログボックスの［相関］を選択して［OK］ボタンをクリックします。すると［相関］ダイアログボックスが表示されるので、［先頭行をラベルとして使用］にチェックを入れます。［入力範囲］に B 列のエネルギーから E 列の売上金額までを選択し、［出力先］で好きな出力先のセルを選択して［OK］をクリックすると、［図 3-3-24］相関行列が表示されるはずです。

カラースケールを使ってわかりやすくするためには、相関行列のセル部分を選択して、ヒートマップの部分を参考に、同じくヒートマップ化することができます。

➲ 連続変数の相関行列［図3-3-24］

	A	B	C	D	E	F	G	H	I	J	K	L
1	商品ID	エネルギー(kcal)	商品価格(円)	売上個数(個)	売上金額(円)							
2	ID_001	83	144	66	9504		Q. 商品価格と売上個数の相関係数を求めてみよう					
3	ID_002	81	182	147	26754		相関係数 =	-0.5426				
4	ID_003	109	250	57	14250							
5	ID_004	82	170	42	7140							
6	ID_005	135	152	57	8664		Q. 連続変数の相関行列を作成してみよう					
7	ID_006	96	188	75	14100			エネルギー(kcal)	商品価格(円)	売上個数(個)	売上金額(円)	
8	ID_007	94	155	81	12555		エネルギー(kcal)	1				
9	ID_008	110	321	153	49113		商品価格(円)	-0.068	1			
10	ID_009	128	179	99	17721		売上個数(個)	-0.009	-0.543	1		
11	ID_010	87	234	42	9828		売上金額(円)	-0.058	0.110	0.734	1	
12	ID_011	102	180	75	13500							
13	ID_012	104	130	102	13260							
14	ID_013	86	180	63	11340							
15	ID_014	94	156	78	12168							
16	ID_015	131	196	45	8820							

これでグッと相関行列が見やすくなりました。やはり商品価格と売上個数の負の相関は高いですが、その逆に、売上個数と売上金額の相関が非常に高いです。常識的に考えると予想通りかもしれませんが、やはり売上個数が金額に対して強い相関関係がありそうですね。そして、エネルギー（カロリー）や商品単価は、売上金額にはあまり相関がなさそうですね。商品のエネルギーはたしかに売上にはそこまで影響はなさそうなので、ある意味予想通りの結果が出ていると考えることができそうでしょうか。

ここまでで、さまざまな記述統計やデータの可視化について取り上げてきました。まだ紹介しきれていないことも多いですが、これだけでもデータの傾向、つまり現実における現状の傾向をある程度把握できます。なお、今回は紙幅の関係から、統計学に関してはあまり深くは取り上げませんでしたが、推計統計と呼ばれる「手元のデータから、そのデータのもととなる母集団全体レベルのことまで適切に把握しよう」というトピックも非常に重要です。特に「仮説検定」や「回帰分析」などは、実務でも非常によく使われるのでぜひ調べてみてください。ただし理論的に押さえておかねばならないことは比較的多いでしょう。以下の「ステップアップにつながるトピック」にて、仮説検定や回帰分析に必要なトピックをリストアップしておきます。

■ ここで学んだ重要トピック

- 要約統計量
- 平均値、中央値、分散、標準偏差、最大値、最小値
- データの可視化
- ヒストグラム、棒グラフ、ヒートマップ、散布図、相関行列（相関係数）

■ ステップアップにつながるトピック

- ピボットテーブル
- 確率分布（正規分布、二項分布、ポアソン分布 など）
- 仮説検定（t 検定、カイ 2 乗検定 など）
- データの前処理（ダミー変数、欠損値の補完、外れ値の考慮 など）

Chapter 4

線形回帰モデルで需要予測を立てる

Section

01 販売数の需要予測により発注精度を向上しよう

可視化や記述統計を通じて、データをきちんと理解するということがイメージついたわね。ここからは、データの理解に基づいた、より高度な技術を具体的なビジネス課題に適用していきましょう。

そうですね、最近よく「AI 活用」とかいわれていますけど、やはりいまいちピンとこないので、しっかり学びたいです！

その心意気よ！ 学びたいトピックはたくさんあるけど、まずは「教師あり学習」と呼ばれる、実務でよく使う手法から見ていきましょう。

えーと、回帰問題と分類問題に分けられるんでしたよね……。でもどうやってビジネス課題に落とし込めばいいかさっぱりです。

ここでは、販売傾向をデータから学習して、店舗業務における発注効率を向上させるという飲食店のケースを通じて、まずは教師あり学習の回帰問題を実務に落とし込んでみましょう。

ここで学ぶこと

- ☑ 教師あり学習の基本的な仕組み
- ☑ 目的変数や特徴量といった基本的なトピック
- ☑ 線形回帰モデルを実務で活用するための考え方

とある飲食店の課題を考えてみよう

とある飲食店Bのケースを考えてみましょう[※1]。B店は開業してから数年がたち、今では安定して売上と利益が出ている状態で、お店も繁盛しています。開業時のオーナーが店長として業務をこなしていましたが、さらなる拡大のため、2～3店舗と増やしていきたいと考えています。しかしいくつか問題があることに気がつきました。

その1つは、今までは店長の「経験と勘」で店舗を運営していたことです。たとえば料理のレシピなどは、ほかの料理人もいたことから、レシピを作成して業務の標準化（マニュアル化）ができていました。しかし、主要な業務の1つである**商品商材の発注業務については標準化ができていません**でした。これまでは店長が開業当初から培ってきた感覚に頼りつつ、POSデータ[※2]を見ながらなんとなく発注をしていましたが、それでもなんとか成り立っていました。しかし、過去を振り返ると、**意外と廃棄商品があったり、お客さんがオーダーした商品が存在していなかったり（機会損失が起こっていた）といったこと**がよくありました。

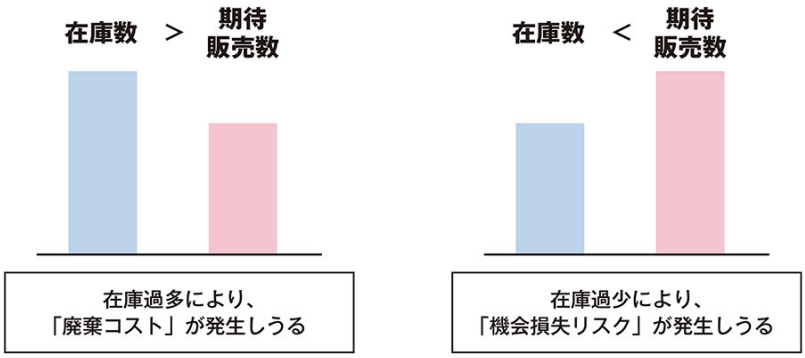
今後、店舗数を増やしていくのであれば、販売傾向や発注量の感覚がない新たな店長と店員で店舗運営をしていく必要があるため、次のような業務改善を行うことにしました。

➲ 業務改善施策［図 4-1-1］

1. 廃棄数や機会損失数といったデータを収集しつつ、発注業務を適切に管理するようなシステムを導入する
2. データに基づいてどの程度発注すればよいかを事前に提案する

（※1）飲食店に限らず、同様の課題がありそうな小売店などの想定でもよいでしょう。
（※2）POS：Point of Sales。販売時点情報管理（販売データの管理システム）

➲ 発注した在庫数と期待される販売数にずれが生じてしまう［図 4-1-2］

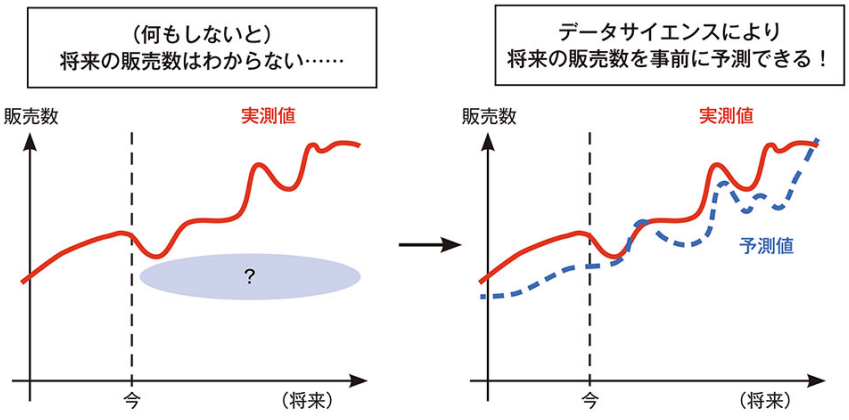


［図 4-1-1］の 1 に関しては、簡単な発注管理システムを導入し、商品を廃棄するタイミングやオーダーに対して商品が提供できなかったタイミングで、システムに入力してもらいます。これにより今後の廃棄数や機会損失数をデータ化することで対応できそうです※3。

2 に関しては、これまでの将来の販売数が明示的にはわからない状態から、**POS データをもとに、データサイエンスの機械学習の技術を使って、将来の販売数を事前に予測することで実現**を目指すことにしました。

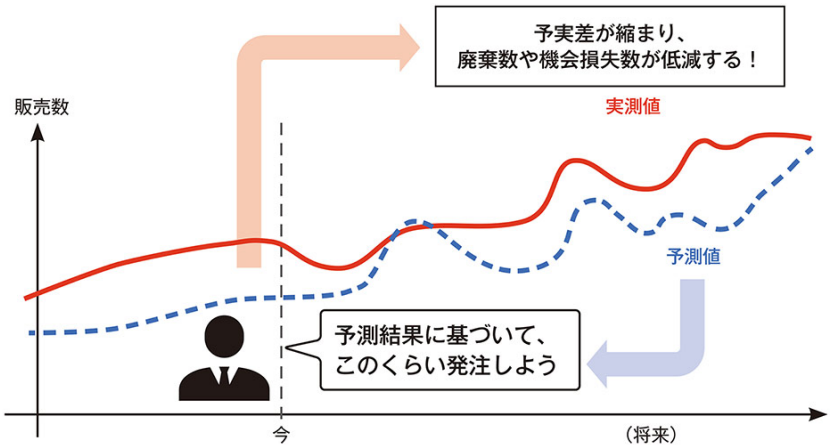
将来の販売数をできるだけ正確に予測できれば、その予測結果に基づいて発注することで、予実差のズレが縮まり、廃棄数や機会損失数といった KPI が改善していくはずです。

➲ データサイエンス技術により、将来の販売数を予測する［図 4-1-3］



（※ 3）どうシステム化するかに関しては、本書の内容からは外れてしまうので取り扱いません。

➲ 予測に基づいた発注により、廃棄数や機会損失数を減らしていきたい［図 4-1-4］



データサイエンスで解くための問題設定

それでは「販売数を事前に予測」するための問題設定をもう少し具体的に考えて、データサイエンスで適切に解ける状態にしておきましょう。今回のケースでは、大きく以下の論点は決めておく必要がありそうです。

➲ 今回のケースにおける論点［図 4-1-5］

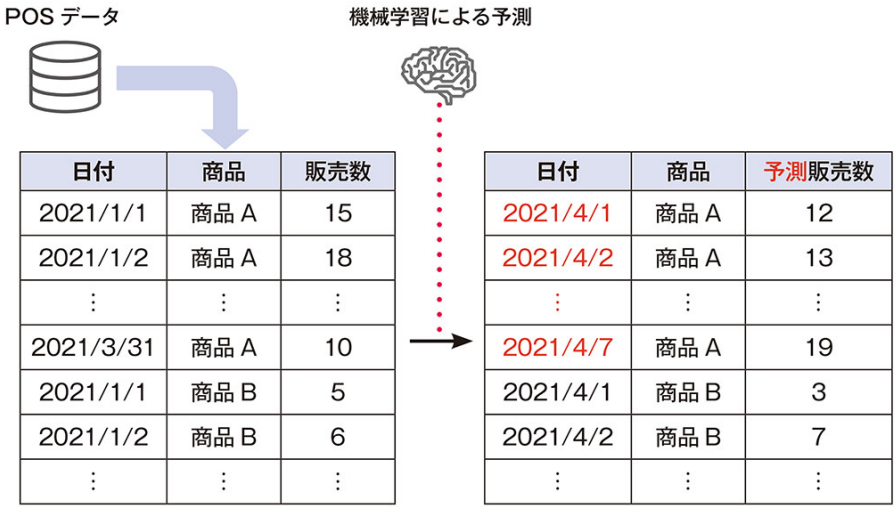
1.「将来」とは、いつからいつまでの期間とするのか？
2. どの粒度で販売数を予測するのか？

1つ目は発注業務との兼ね合いになりますが、今回はシンプルに「ある日に、次の日の発注をする業務である」と設定しましょう。その場合**「ある日において、明日の販売数を予測する」**という問題設定となりそうです。たとえば「ある日において、次の1週間の販売数分を一気に発注する必要がある」という業務であれば「ある日から将来の1週間分の販売数を予測する」という問題設定とすればよいでしょう。

2つ目の「どの粒度で販売数を予測するのか」に関しては、必ずしも正解があるわけではありません。そのため実存するデータを眺めつつ、できるだ

け適切に設定する必要があります。たとえばPOSデータであれば、多くは「時間ごとに、各商品がどの程度販売されたか」といったデータが蓄積されているはずです。その場合、そのようなPOSデータを集計し、機械学習（教師あり学習）に学習させ、**「日別・商品別の予測販売数」**を出力させられます。このように、**「予測値をどの粒度（＝どのような行単位）で出力するか？」**を設定しておく必要があります。ちなみに、次の1週間分の販売数を知りたいのであれば、次1週間分の日別の販売数ではなく「1週間の合計販売数」を直接学習・予測させることも可能です。そしてどちらのほうが精度がよいか？というのは実際に構築・検証してみないとわかりません。「日別・商品別の販売数」の粒度も含め、さまざまなパターンで試行錯誤してみるのがベストです。

➲ これまでのPOSデータから、日別・商品別の販売数を予測する［図4-1-6］



日付	商品	販売数
2021/1/1	商品A	15
2021/1/2	商品A	18
⋮	⋮	⋮
2021/3/31	商品A	10
2021/1/1	商品B	5
2021/1/2	商品B	6
⋮	⋮	⋮

日付	商品	予測販売数
2021/4/1	商品A	12
2021/4/2	商品A	13
⋮	⋮	⋮
2021/4/7	商品A	19
2021/4/1	商品B	3
2021/4/2	商品B	7
⋮	⋮	⋮

これで、ある程度データサイエンスにより解ける問題設定となってきました。ただし留意しておきたい点が2つあります。

1つは、今回はあくまで販売数を予測するモデルであるということです。もし仮に発注の単位が「商品」（牛丼、オムライスなど）ではなく「商材」（牛肉、卵など）であれば、商品ごとに必要商材を算出するための情報が必要となります。ただし現実的には、商材そのものを予測するのは難しいので、POSデー

タからわかる販売数を学習・予測して、その数値と商品ごとの必要商材数の情報をもとに商材数を算出することが多いです。またもし発注商材にロット（じゃがいもは20個セットで発注する必要あり、など）があれば、その点も加味する必要があります。今回はそこまで加味すると複雑になるため割愛しますが、実務上はそういった制約も把握、加味しながら進めていく必要があるでしょう。

2つ目の留意点は、基本的にPOSデータには機会損失の情報がないということです。つまり、あくまでPOSデータは「販売できた個数」しか記録されないため、注文しようと思ったけれど欠品していたので頼まなかった、という機会損失数は加味されていません。本来であればPOSの販売数に加えて、そのような機会損失がどの程度起きていたかといったデータも収集・加味したほうが正確です。しかしそのようなデータは収集も難しく、蓄積されていないことが多いので、今回はPOSデータのみでモデルを構築していきます。ただ、上記のような留意点があり、今後機会損失のデータも蓄積していくべきだと提案する必要がある、という部分を頭の片隅に置いておくとよいでしょう。

さて、次Sectionからは、実際に販売数を学習・予測するための「教師あり学習」を構築していくにあたり必要な知識を学んでいきましょう。数学的な話がどうしても出てきますが、できるだけ平易に説明していくので、一緒に学んでいきましょう。

ここまで問題設定をする中でいくつかの留意点を挙げましたが、重要なトピックとして「学習」と「予測」という単語が出てきましたね。教師あり学習では、この学習と予測の概念をしっかり理解することが重要です。次のSectionでは、教師あり学習の全体像を学びながら、「学習・予測とは何か？」を理解していきましょう。

Section

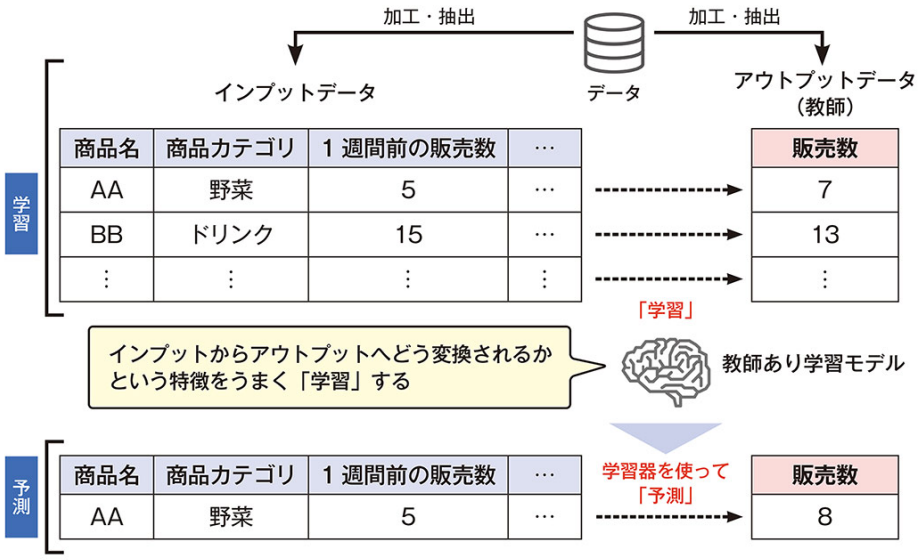
02 教師あり学習（回帰問題）の概要

教師あり学習モデルによる「学習」

それでは具体的な技術論を学んでいきましょう。第2章で紹介した教師あり学習を、本ケースの販売数予測の視点で細かく見ていきます。

教師あり学習は、基本的に「学習」と「予測」のパートに分かれています。まずは「学習」フェーズを押さえましょう。

➲ 教師あり学習の全体像（学習と予測）［図 4-2-1］



学習フェーズの重要なポイントは、まず**インプットデータとアウトプットデータを定義し収集するという点**です。今回の例では、過去のデータをベースとして次のようにデータを定義し、収集・加工します。

➲ インプットデータとアウトプットデータの定義 [図 4-2-2]

・インプットデータ：商品ごと・日付ごとのさまざまな情報

例）商品カテゴリ、時間、日、前日の販売数、1 週間前の販売数……など

・アウトプットデータ：商品ごと・日付ごとの販売数

たとえば、2021/1/1 ～ 3/31 の POS データがあったとしたら、2021/1/8 の商品 A に関しては次のように定義できます。

➲ データ定義と収集・加工の例 [図 4-2-3]

・インプットデータ：

○商品カテゴリ：商品 A の商品カテゴリ

○日付・時間：8 日、2021/1/8 の曜日（金曜日）

○前日の販売数：商品 A の 2021/1/7 の販売数

○ 1 週間前の販売数：商品 A の 2021/1/1 の販売数

・アウトプットデータ：

○商品 A の 2021/1/8 の販売数

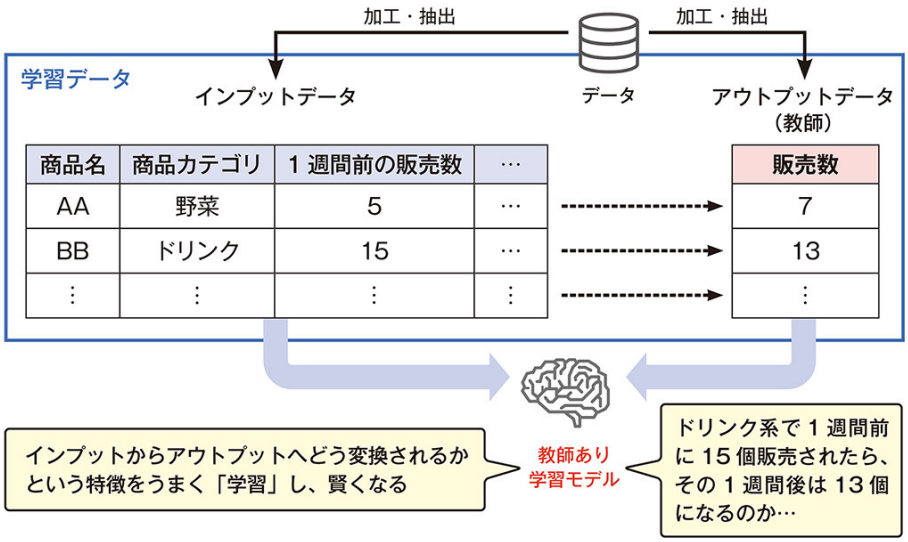
➲ 学習データのイメージ [図 4-2-4]

インプット							アウトプット
商品 ID	商品カテゴリ	日付	曜日	日	前日の販売数	1 週間前の販売数	販売数
id_8280607	野菜	2021/2/1	月曜	1	2	4	4
id_8280607	野菜	2021/2/2	火曜	2	4	11	6
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
id_8280607	野菜	2021/3/28	日曜	28	13	6	4
id_5029823	米類	2021/2/1	月曜	1	4	19	15
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
id_5029823	米類	2021/3/28	日曜	28	21	20	50
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

すべての商品・すべての日付で同様の計算をすることで、学習データを集めることができ、［図 4-2-5］のような学習データイメージとなります。

そして、インプットデータとアウトプットデータを定義した学習データをもとに、**教師あり学習モデルが「インプットからアウトプットへどう変換されるか？」という特徴をうまく学習**します。

➲ モデルはインプットデータとアウトプットデータをもとに学習する［図 4-2-5］

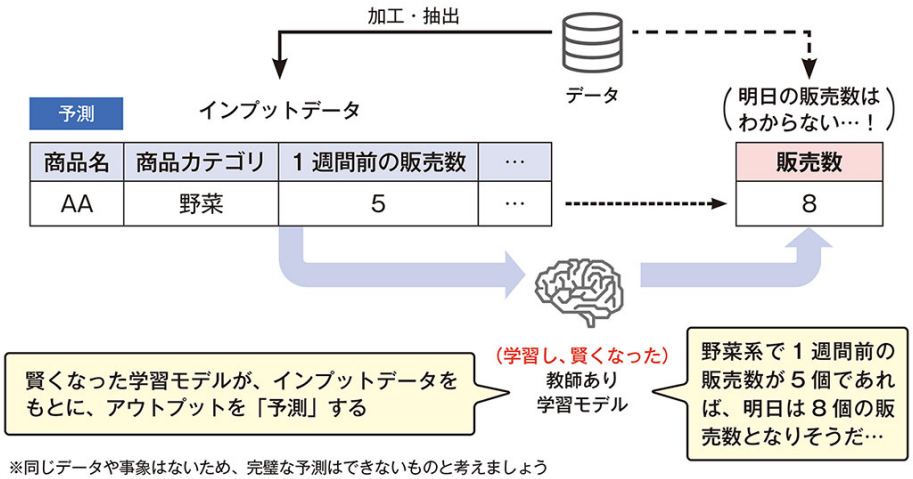


学習したモデルを用いて「予測」する

学習によって賢くなったモデルによって、予測ができます。流れとしては、学習データとは別のデータを、すでに学習したモデルに入れることにより、予測値を算出できます。その際に、学習したモデルが賢くなっていればいるほど、より正確な予測値を算出可能です。今回のケースで考えれば、ある商品・ある日に関して、その次の日の販売数をできるだけ正確に予測することができる、といったイメージです。

とはいえ、学習データとまったく同じデータや現象が起きるということはないはずです。仮に同じ商品だとしても、これまでと明日以降で天気や人の流れなど、何かしらの違いは多少なりともあるでしょう。そのため**完璧に（100% の）精度で予測できるわけではない、ということに注意しましょう**。

➲ 学習したモデルを用いて、未知なるインプットデータから予測する［図 4-2-6］



※同じデータや事象はないため、完璧な予測はできないものと考えましょう

モデルはどう学習するのか？

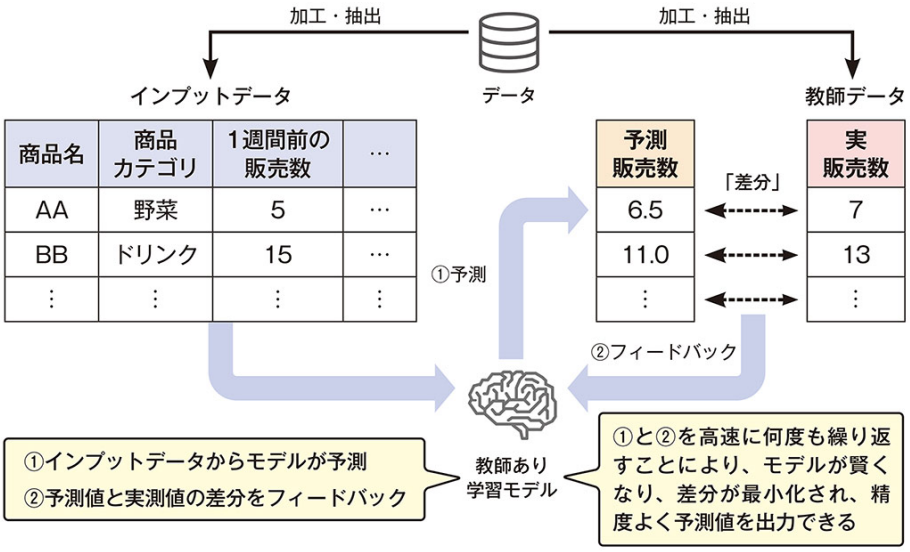
ここまでで学習と予測の概要を説明しましたが、**実際にどのようにモデルが学習しているか**を、もう少し紐解いてみましょう。

インプットデータとアウトプットデータをモデルに入れたからといって、すぐにモデルが学習し賢くなるとは限りません。初期状態では、インプットデータをモデルに入れても、モデルはまったく見当違いな予測値を算出します。その場合、アウトプットデータである実測値とは大きく乖離しているはずです※4。したがって、その予測値と実測値の差分を、より小さくするようなフィードバックをモデルにかけます。すると、モデルは少し賢くなり、次はより実測値に近しい予測値を算出するようになります。

このサイクルを、人間が計算するよりも何十倍何百倍も高速に何回も繰り返すことで、予測値と実測値の差分が最小化され、モデルが賢く学習された状態となります。そうすると、モデルが精度の高い予測値を算出できるようになります。

（※4）もう少し詳しくいうと「たまたま実測値と近しい予測値を取っているデータもあれば、大きく乖離しているデータもいろいろあるが、データ全体で平均的に考えると、初期状態では予測値と実測値に乖離がある」ということになります。

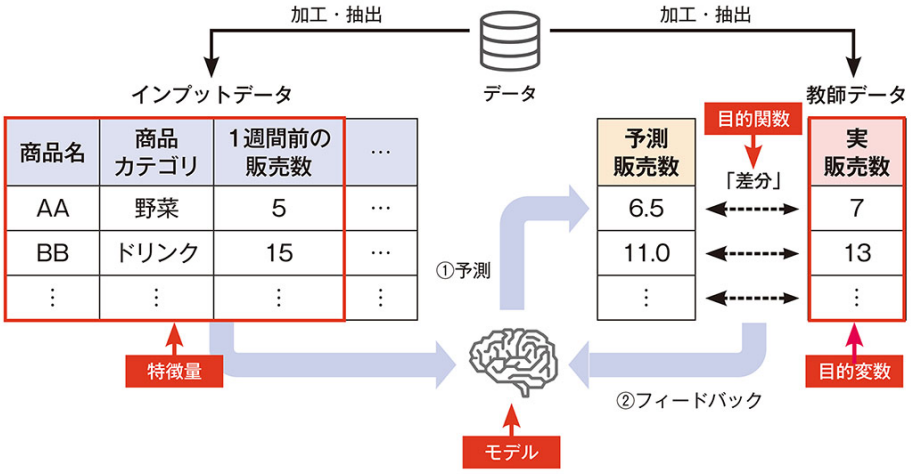
➲ 実測値と予測値の差分を最小化するようにモデルにフィードバックする［図4-2-7］



これだけは覚えておこう！教師あり学習の用語

ここまでは、できるだけ平易な言葉で説明してきましたが、ある程度覚えておいたほうがよい専門用語があります。［図4-2-8］に示した4つの用語は、重要なため覚えておきましょう。

➲ 覚えておきたい4つのデータサイエンス用語［図4-2-8］



・**目的変数：学習・予測対象とするデータ**※5
　○今回の例では「商品ごと・日付ごとの販売数」に相当します。
・**特徴量：インプットデータとして定義する、目的変数の特徴を定量化した数値。データ構造としては、インプットデータの各カラムのこと**※6
　○今回の例では、商品カテゴリ・日付・前日の販売数・1週間前の販売数といった部分に相当します。
・**モデル：インプットである特徴量からアウトプットである目的変数への変換器**
　○実際にはさまざまな機械学習や統計学のモデルの種類が存在し、それらから適切と考えられるモデルを選択していきます。学習により、このモデルが最適な形へと変換されます。
・**目的関数：特徴量をモデルへ入れた際に算出される予測値と、目的変数である実測値の差分**※7
　○実際にはさまざまな差分の計算方法（定義）が存在し、そのつど適切な目的関数を選択します。この目的関数を最小化するように、モデルが学習します。

これらの用語は、データサイエンスに関わる場面でよく出現するので、必ず覚えておきましょう。また本書でも、今後はこれらの用語を使用し、理解に齟齬がないように進めていきます。

精度を上げるための3つのアプローチ

最後に、少し実務的な話をして次のSectionに進みます。皆さんが今後、教師あり学習モデルを構築していく際に気になる点として、どうしたら精度のよいモデルが手に入るだろうか？という疑問があると思います。もちろんさまざまな技術的工夫を施していくことで、精度は徐々に改善していくのですが、わかりやすく大きなアプローチとしてまとめると、［図4-2-9］の3つが挙げられます。

（※5）「ターゲット変数」という場合もあります。また、統計学の分野では、しばしば「被説明変数」と呼ばれることもあります。
（※6）「被説明変数」と対応する形で、「説明変数」と呼ばれることもあります。
（※7）「誤差関数」や「損失関数」と呼ばれることもあります。

➲ モデルの精度を高めるアプローチ［図 4-2-9］

① データ量を増やす
② 特徴量を増やす
③ モデルを複雑にする

1つ目は**データ量を増やす**ことです。過去1週間分しかないより、過去1年間分あったほうが、より多くの事象の傾向をつかめます。これはできるだけ早い段階から、根気よくデータを蓄積していくしかありません。ただし、ただデータを貯めるだけでは「手段」で終わってしまいます。どのような「目的」を果たしたいか、そのためのモデル構築としてどのようなデータが必要か、と一気通貫した形でデータを蓄積していきましょう。

2つ目は、**特徴量を増やす**ことです。これも同様に、商品カテゴリしかないよりも、日付や1週間前の販売数といったさまざまな特徴量があったほうが精度がよくなりそうですよね。特徴量を増やす1つの方法としては、できるだけ多くの種類のデータを集めておく、となります。一方で、今手元にあるデータからいかに特徴量を生成していくか、という考えで特徴量を増やす方法を、**特徴量生成（Feature Engineering）**といいます。今回の例でいえば、手元にあるPOSデータを加工することで、「1週間前の販売数・過去1週間の平均販売数・過去1か月の平均販売数……」と、さまざまな特徴量を生成できます。このように手元のデータ種類を増やしつつ、Feature Engineeringにより特徴量を増やすことで、精度を向上させます。このときには、**目的変数に影響を及ぼしていそうな特徴量は何か？を考えるためのビジネスドメイン知識と、それを適切にデータ加工するエンジニアリングスキルの両方を組み合わせる必要**があります。

3つ目は、**モデルを複雑にする**ことです。機械学習や統計学に存在するさまざまなモデルから適切と思われるものを選択し、時に複雑なモデルにすることで精度を向上させます[※8]。本書では次Sectionにて、モデルの中でも一番基本的かつ重要な「線形回帰モデル」を学びます。またこの後の章では、より複雑なモデルであるディープラーニングモデルも紹介します。

（※8）もちろん、必ずモデルを複雑にすればよいわけではなく、ケースバイケースで異なってくるので、使用するデータごとに適切なモデリングをする必要があります（ときには比較的シンプルなモデルでよいという場合もあります）。また特徴量も同様に増やしすぎると精度が下がるケースもあるので、注意が必要です。

Section

03 回帰問題の基本手法「線形回帰モデル」

単回帰分析の概念を理解する

さて、このSectionでは、前のSection 02で出てきた「モデル」を詳しく紐解いていきましょう。前述の通り、モデルと一言にいっても、機械学習や統計学のモデルは非常に多く存在します。今回の教師あり学習の回帰問題で使えそうな主要なモデルとしては、以下のようなものが挙げられます。

➲ 回帰問題で使える主なモデル［図 4-3-1］

- 線形回帰モデル（単回帰分析、重回帰分析）
- Ridge 回帰、Lasso 回帰、Elastic Net（線形回帰モデルの派生系モデル）
- 決定木
- ランダムフォレスト（XGBoost、LightGBM などの派生系モデルも多数存在）
- SVM（Support Vector Machine）
- ニューラルネットワーク、ディープラーニング
- 時系列モデル（AR モデル、MA モデル、ARIMA モデルなど）

これらすべてのモデルを紹介したいところですが、理解するための技術的難易度や紙幅の関係から、今回は1つ目に挙げた「線形回帰モデル」を紹介します。実務では、さまざまなモデルを利用していくこととなりますが、線形回帰モデルでの概要をつかめれば、そのほかのモデルでも（モデルは複雑になってくるが）本質的には同じことをやっていると思って大丈夫です。興味があれば、ぜひいろいろと調べてみてください※9。

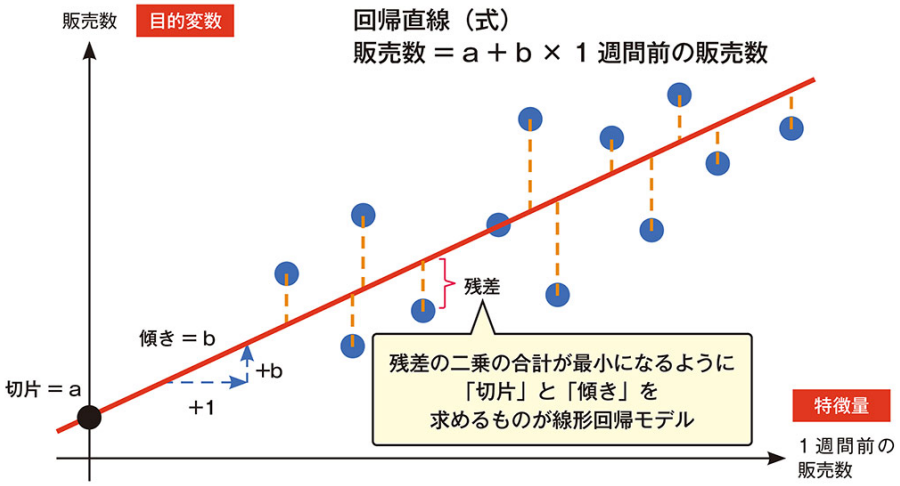
それでは、線形回帰モデルの概念を理解していきましょう。まずは特徴量が1つの場合である「単回帰分析」を取り上げながら、線形回帰モデルを順

（※9）実務的にはランダムフォレストといわれるモデルの派生であるLightGBMというモデルが、比較的どのようなデータに対しても精度が高く、よく使われます。また画像に対しては前章で説明したようにディープラーニング系のモデルが使われます。

に説明していきます。

まず端的に、線形回帰モデルとは「**特徴量と目的変数の関係性を最もよく表している直線式を求めること**」であるといえます。今回のケースで考えましょう。特徴量は何でもよいのですが、たとえば1週間前の販売数から（そこから1週間後の日付の）販売数を学習・予測したいとします。1週間前の販売数が多いほどそこから1週間後の販売数も基本的には多いはずなので、［図4-3-2］のようなデータとなりそうです。

➲ 線形回帰モデルはデータを直線で表現するモデル［図4-3-2］



線形回帰モデルによって直線式を求めるとすると、以下のように表せます。

➲ 特徴量と目的関数の関係を直線式で表す［図4-3-3］

販売数＝切片a＋傾きb × 1週間前の販売数

単回帰分析における「学習」と「予測」

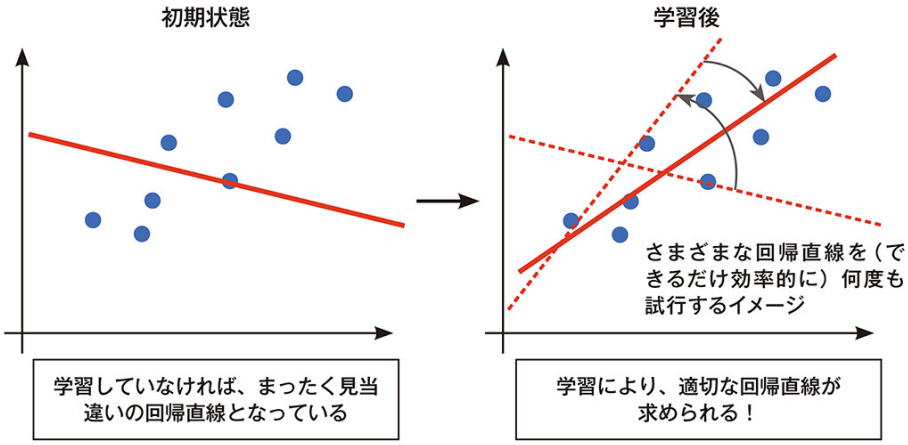
そして続いて重要なポイントとして、このａとｂは何かしらの数値を代表するための記号にすぎないのですが、どのような値でも取りえます。したがって、このａとｂが決まらない限りは直線式が定まりません。線形回帰モデルにおける「学習」とは、「**すべてのデータ点に関するデータと直線上の点の距離である『残差』の二乗の合計が最小になるように、『切片 a』と『傾き b』を求める**」と定義されています。

残差が小さくなるほど、データと直線上の点が近づくので、直線がよりデータを表しそうですよね。**何も学習していない初期状態であれば、回帰直線の切片や傾きはテキトウな値となっています**が、さまざまな回帰直線を試す、すなわち［図 4-3-2］における「切片」や「傾き」をいろいろと変えてみることによって、**一番適切な切片や傾きが見つかり、最適な回帰直線を求められます**。

つまり、この**「残差の二乗の合計」は「目的関数」**となっています。目的関数とは、予測値と実測値の差分を表す式でしたね。まさに残差の二乗の合計は目的関数としての１つの式であり、この目的関数を最小化する必要があります。つまり、回帰直線の傾きや切片がテキトウな値となっている何も学習していない初期状態は、目的関数が悪い状態であると考えられます。そこから、「もっと目的関数を改善してほしい」と回帰直線であるモデルにフィードバックすることで、切片や傾きを改善させていきます。目的関数を最小化するには、さまざまな切片や傾きの組み合わせを考える必要がありますが、その計算には多くの時間がかかります。そのため実際には何かしら効率的に探索することで計算時間を短縮します。その例として、「勾配降下法」といった方法論などが挙げられますが、本書では紙幅の関係上その説明は省きます。最終的に、**目的関数が最小となるような切片や傾きを得ることで、それがデータに最もフィットした最適な回帰直線である**、ということができます。なお、次ページの［図 4-3-4］のようにいろいろと試行錯誤をしなくとも、最適な回帰直線の切片や傾きはデータが与えられた際に解析的に（手計算により）求められます。ただし、線形回帰モデルではない複雑なモデルとなってくると、解析的に求められないことがほとんどです。そのため、今回はあえてほ

かのモデルでも通用するような、繰り返し処理によりモデルを最適化するという考え方を紹介しています。

➲「学習」を繰り返すことにより適切な回帰直線が求められる［図4-3-4］

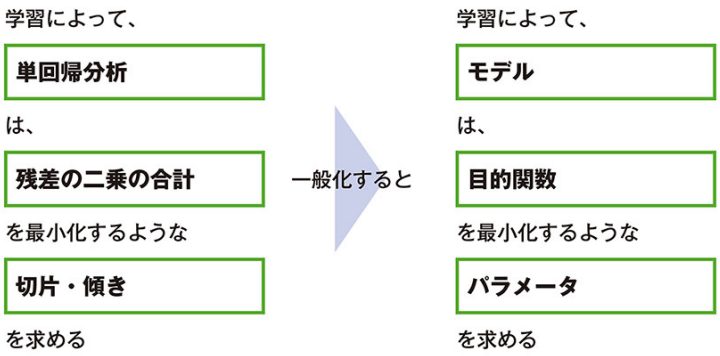


この［図4-3-4］では2回の修正で最適な回帰直線となっているように描画していますが、実際は必ずしも2回というわけではないことに留意してください。特に、モデルが線形回帰モデルではないような複雑な場合、非常に多くの繰り返し試行が必要となります。

ここまでの単回帰分析の話を一般化し、汎用的な知識としましょう。回帰直線というのは、**「データを直線で表すモデル」**であるといえます。そしてデータを与えることにより、回帰直線は（残差の二乗の合計である）目的関数を最小化します。最小化することで最適な切片や傾きを求められ、回帰直線をデータに対してフィット＝学習させられます。

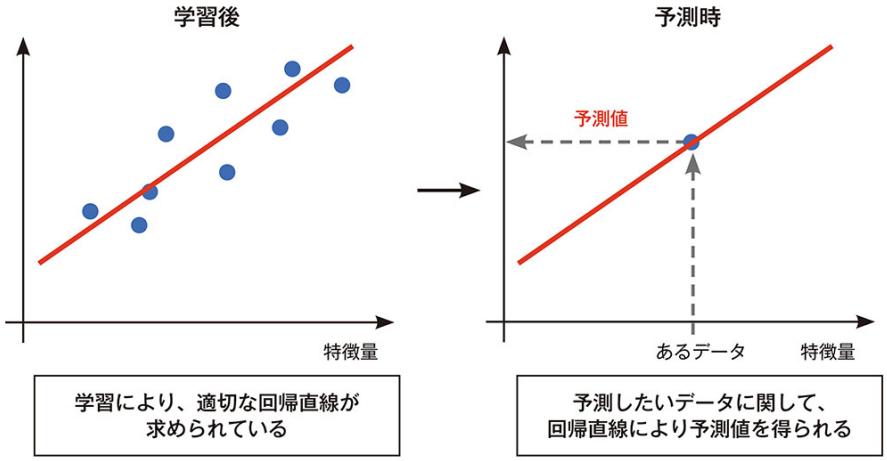
このような、目的関数を最小化することで得たい切片や傾きといったモデルの形状を決める変数を**「パラメータ」**といいます。つまり**「学習によって、モデルは目的関数を最小化するパラメータを求める」**といえます。

➲ モデル・目的関数・パラメータの関係性［図 4-3-5］



学習により最適な回帰直線を得られれば、あとは予測ができます。学習には用いていない新たなデータがきた際に、そのデータから特徴量を計算します。あとは、学習した回帰直線にインプットすることで、回帰直線の式から計算されて予測値を算出できます。

➲ 学習により得られた回帰直線から、予測することができる［図 4-3-6］

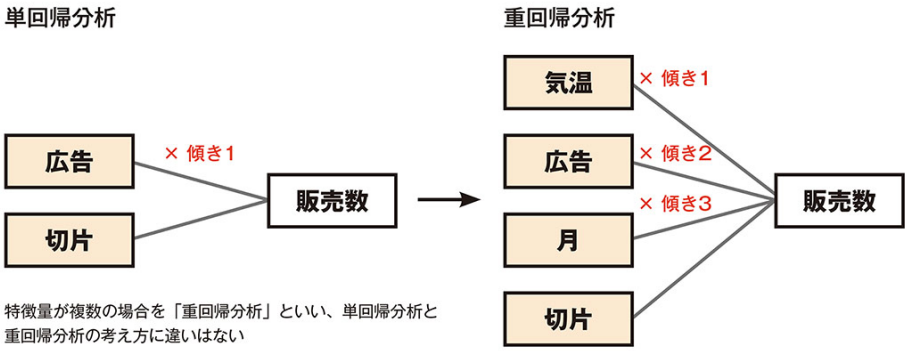


重回帰分析により「複数 対 1」変数の関係性を理解する

前の Section では、1 つの特徴量 対 目的変数という単回帰分析を紹介しましたが、実際に単回帰分析を行うケースは多くありません。当たり前ですが、ある目的変数を説明するような事象（変数）が 1 つであるということはまずないからです。またできるだけ精度を上げるために特徴量を増やしたいはずです。

特徴量が 2 つ以上ある場合は「重回帰分析」といわれますが、**単回帰分析も重回帰分析も基本的な考え方に違いはなく、単純に特徴量の数が増えているだけ**、と捉えてください。

➲ 単回帰分析と重回帰分析は基本的に同じ考えである［図 4-3-7］



また重回帰分析となると、「複数の特徴量 vs 1 つの目的変数」という構図になるので、可視化しようとしても単回帰分析のように散布図で表現できません。そこで、［図 4-3-7］の右側のような重回帰分析を、下記のように定式化します。

➲ 重回帰分析の式［図 4-3-8］

$$\text{目的変数} = \text{切片 a} + \text{傾き b1} \times \text{特徴量 1} + \text{傾き b2} \times \text{特徴量 2} + \text{傾き b3} \times \text{特徴量 3} \ldots$$

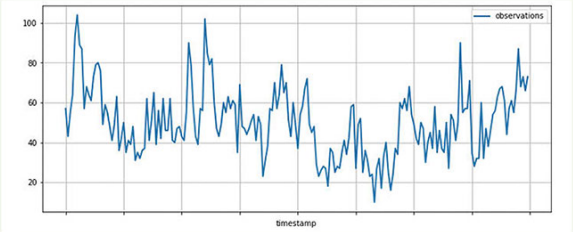
また重回帰分析の場合も単回帰分析と同様に、与えられたデータから、各特徴量の傾きおよび１つの切片（パラメータ）を動かしていき、予測値と実測値の差分となる目的関数を最小化していきます。最終的に最小化された段階が学習できている状態、すなわちパラメータである各傾きと切片が最適な値となっており、適切な回帰直線が得られているという流れとなります。

ほかにもたくさんのモデルが存在するけど、流れとしてはすべてのモデルで**「学習によって、選択したモデルが、目的関数を最小化するようなパラメータを求めることにより、モデルが最適化され、高い精度で予測が可能となる」**となるわね。

Tips　時系列モデルについて

今回は将来の販売数を学習・予測するために機械学習の教師あり学習を利用しますが、時点別の販売数といったデータを分析する際には、統計学の技術である「時系列モデル」を利用するケースもあります。機械学習のモデルと時系列モデルのどちらがよいかはデータの傾向やデータ量によって異なりますが、基本的にはどちらを使っても間違いにはなりません。

今回は機械学習の教師あり学習モデルの基本的な部分だけを押さえておくこととしますが、教師あり学習モデルの基本部分を理解して、それをビジネス適用するイメージがつけば、あとは時系列モデルだろうが複雑なモデルだろうが本質的には同じです。よってビジネス現場への活用という観点からは、十分であるとも考えています（もちろん、データサイエンティストになるのなら、より幅広い技術を押さえておく必要があります）。



時系列データのイメージ（横軸：時間、縦軸：対象変数）

Section

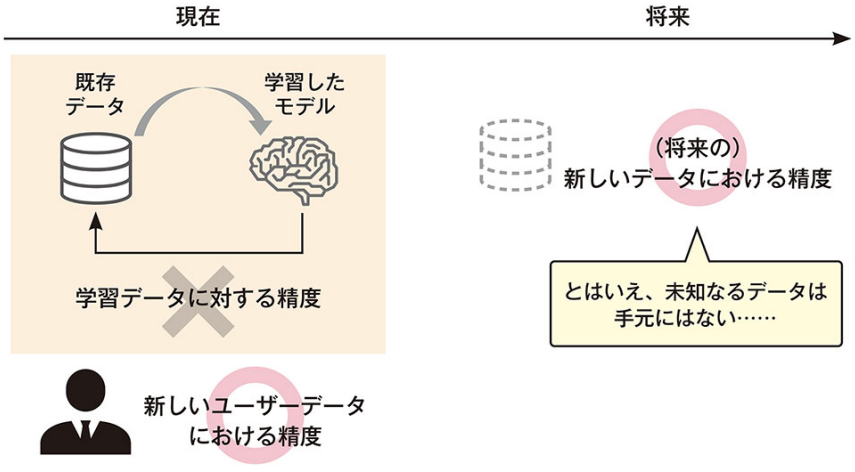
04 予測モデルの精度を評価するための評価指標

精度評価指標を考える

Section03では線形回帰モデルを例に、モデルがどう学習し、どう予測するか？を学びました。人間が多くを考えずともモデルが予測してくれるという恩恵は非常に大きいですが、そもそも**そのモデルがどのくらいの精度を持っているのか？**というのは検証する必要があります。目的関数と同じ考え方になりますが、精度というのは基本的には**「予測値と実測値の差分が小さければ小さいほどよい」**という考えを前提とします。

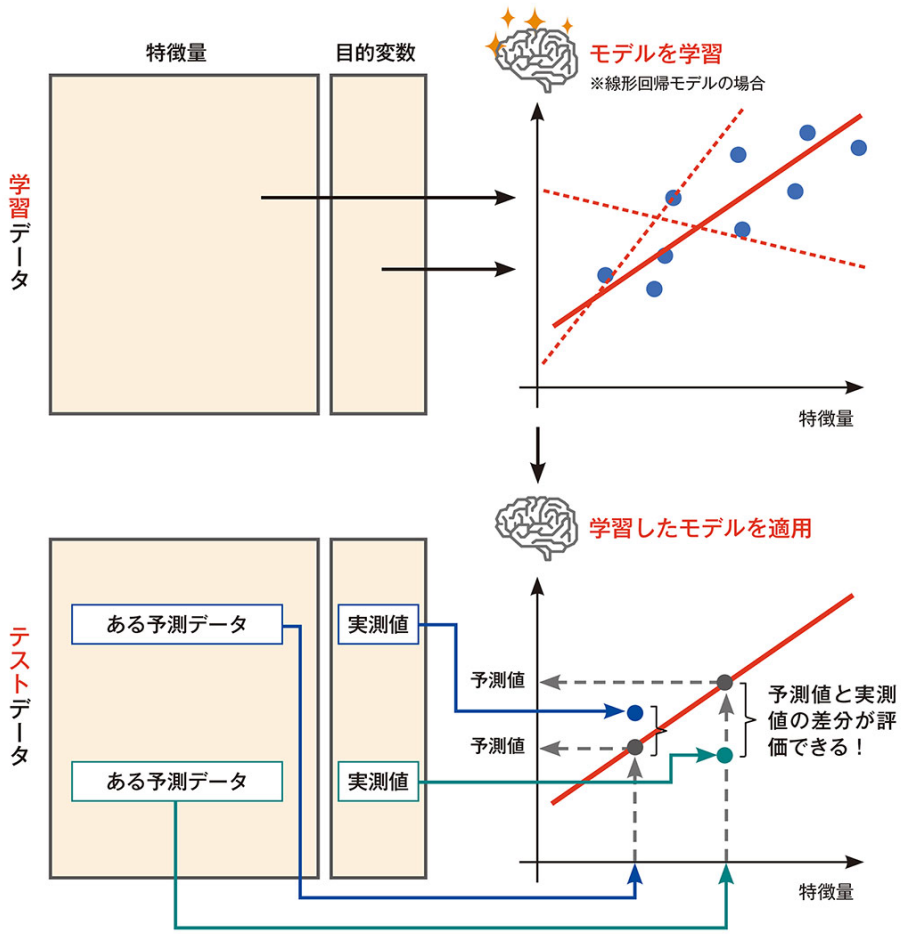
加えて、機械学習においては**「手元のデータではなく、新たな未知なるデータにおける精度を知りたい」**という重要な概念があります。今回の例である販売数を求めるケースでも、これまでの需要は当然わかっているわけで、あくまで過去ではなく将来の需要を精度よく予測したいはずです。とはいえ、将来の情報や未知のユーザーといったデータは当たり前ですが存在しないので、わかりません。

➲ 機械学習では、未知なる新たなデータにおける精度を気にしている［図 4-4-1］



そこで、「なんとかして手元のデータを使いつつ、できるだけ未知なるデータでの精度を検証できるようにしよう」という考え方を導入します。それは、**「手元のデータを、モデルを学習するための学習データと、精度を検証するためのテストデータに分割する」**というアプローチです。これにより、モデルが学習したデータ以外のデータ、すなわち“擬似的に”未知なる新たなデータで精度をチェックできます。またこのテストデータは**手元にあるデータなので、実測値としての実際の正解の値もわかっており、学習したモデルによる予測値と実測値を比較することにより、実際に精度を計測できます**。

➲ 学習データで得られたモデルをテストデータに適用し、精度を計測する［図 4-4-2］

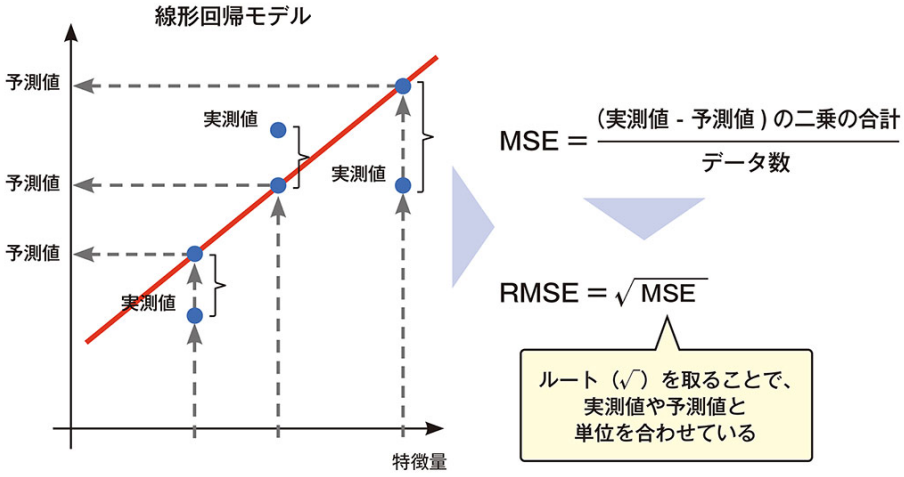


その際に計測する予測値と実測値の差分を評価するための指標を「精度評価指標」といい、いくつもの精度評価指標が存在します。また**教師あり学習の中でも、目的変数として連続数値を取り扱う回帰問題と、カテゴリカル変数を取り扱う分類問題で使える指標は異なります**。そこでまず本章では回帰問題に関して、その中でも有名な指標である「RMSE」と「決定係数」を紹介します[※10]。分類問題における指標は次の第5章で紹介します。

RMSEを理解する

まずはRMSE（Root Mean Squared Error、平均二乗偏差）を紹介します。上述したように、学習データで学習したモデルを予測データに対して適用させます。すると予測データにおける各データの特徴量から、各々の予測値が得られます。各データは（手元のデータなので）実測値も存在しているので、その予測値と実測値を比較します。その際次のように精度を定義します。

➲ MSE、RMSE の定義［図 4-4-3］



まず、MSE（Mean Squared Error）を定義します。MSEは、要するに予実差の二乗の平均値を表しています。ここで「二乗」としているのは、目的関数の際に残差の二乗の合計としたときと同様に、残差のプラスマイナスの影響を受けないようにするためです。

（※10）ほかにも、MAE、MAPEなどさまざまな指標があるので、興味があれば調べてみましょう。ただすべてを覚える必要はありません。何か新たな指標に出会った際に、きちんと調べてその定義を理解することができればよいです。今回紹介するRMSEと決定係数は、覚えておいて損はないでしょう。

しかし、このままだと**二乗のままなので、MSE の場合は単位がもともとの目的変数の単位と揃わなくなってしまいます**。たとえば目的変数が販売数であれば、MSE の単位は「販売数の二乗」となってしまうのです。MSE にルート（平方根）をかけることで二乗を解消すれば、単位が揃い比較的解釈がしやすくなります。したがって、MSE も RMSE も評価指標なのですが、基本的には MSE が使われることはめったになく、どちらかというと RMSE を使うことがほとんどです。

かなり平たく表現するならば、**RMSE とは、「予測値と平均値が平均的にどの程度の乖離があるか」を表している**といえます。もちろんデータによって、完全に予測値と実測値が一致している予測データもあれば、かなり乖離しているデータもあります。RMSE は、平均的に乖離している度合いを示しているのです。つまり、**MSE も RMSE も値が 0 に近づくほど精度がよい**といえます。

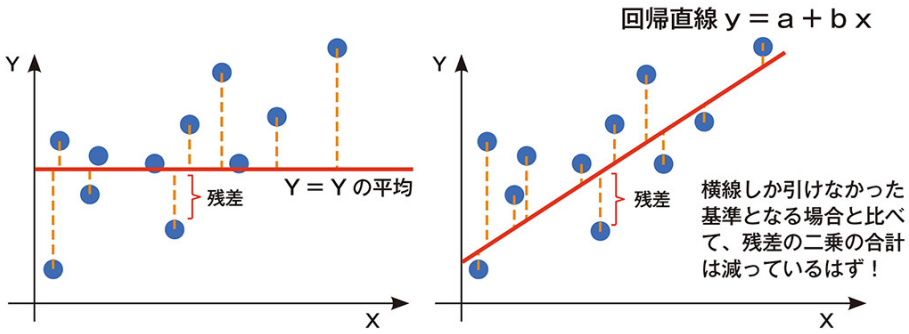
注意点が 1 つあります。それは、MSE も RMSE も予測値と実測値の乖離度合いを示しているので、**0 に近づくほど精度がよい**のですが、どのくらいの値の範囲に収まればよいかという定義はないということです。要するに、目的変数の単位や値の大きさに依存するので、目的変数が 100 前後の値を取るようなデータの場合と 0.1 前後の値を取るような場合で、RMSE の値はまったく変わってきますね。したがって、RMSE を評価する場合は、目的変数の値の大きさを見ながら判断していくことになります。

決定係数を理解する

もう 1 つよく使われる精度評価指標として、「決定係数」という概念があります（R^2、R- スクエアともいわれます）。

［図 4-4-4］の左側を見てください。決定係数では、特徴量 X が与えられていない場合の**「目的変数 Y の平均値（［図 4-4-4］における横線）」を基準（初期値）**とします。そしてすべてのデータに関する Y の平均値とデータ点の「残差の二乗の合計」を基準指標と定義します。

➲ 決定係数は目的変数（Y）の平均値を基準としている［図4-4-4］



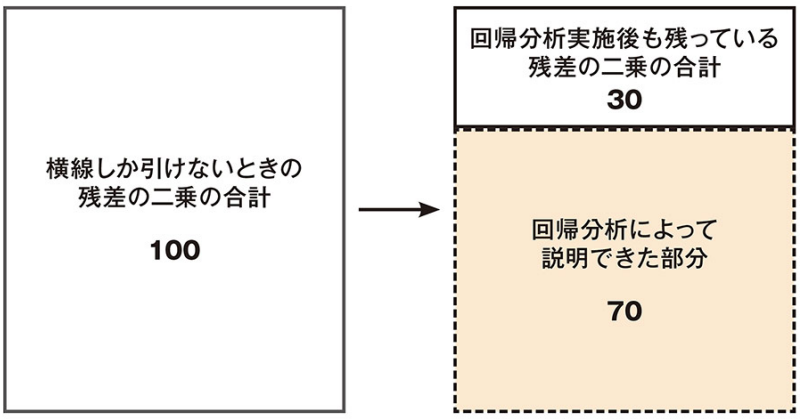
「Y の平均」のみで構成されたモデルを考えた際の、残差の二乗の合計を基準に考える

そのうえで、仮に線形回帰モデルで学習し、回帰直線が引けたならば、その**回帰直線とデータ点の「残差の二乗の合計」はどれだけ減少しているか？を計測しているのが決定係数**です。したがって、Y の平均値と比べて回帰直線のほうが、残差の二乗の合計は基本的に減少しているはずです。

この「Y の平均値」と「回帰直線」を比較することにより、決定係数が計算できます。具体的には次ページの［図 4-4-5］に記載していますが、まずは「Y の平均値」における残差の二乗の合計を 100 とします。そして、線形回帰モデルによって得られた「回帰直線」における残差の二乗の合計が、「Y の平均値」の 100 に比べてどの程度になったかを比較します。たとえば 100 に対して 30 だったとすると、線形回帰モデルによって説明できた部分は 70 といえそうです。この 70（%）が決定係数です。

つまり決定係数とは**「基準となる Y の平均値しか引けなかったときの残差の二乗の合計に比べて、モデルによる予測結果の場合の残差の二乗の合計がどれだけ減少しているか」**を示すことにより、線形回帰モデルによってどれだけ回帰直線をデータに当てはめられるようになったのか？を**0 から 100%の絶対的な指標で表している**のです。

➲ 決定係数の定義式［図4-4-5］



$$\text{決定係数} = 1 - \frac{\text{回帰分析実施後も残っている残差の二乗の合計}}{\text{横線しか引けないときの残差の二乗の合計}} = 0.7$$

つまり決定係数は、**高ければ高いほど（1 に近づくほど）精度がよい**といえます。ただし、必ずこの基準値を超えれば OK というものはありません。もちろん、高い精度が必要な場合は、基本的により高い決定係数が求められます。今回は機械学習モデルにより精度高く予測しようといった場合を想定していますが、精度よりは解釈性が求められる（その分析によりデータからできるだけ示唆を得たい）といった場合であれば多少低い決定係数でも問題ありません。またもし過去から継続的に行っている分析やプロジェクトがあるのであれば、そのような過去の分析結果から比べて相対的にどうか？といった視点で見てみるとよいでしょう。

Section

05 実践：飲食店のPOSデータを活用しよう

練習用ファイル：chap04_demand_forecast / dataset.csv

実践 データの確認

本章最後のSectionでは、実際のデータを用いて、販売数の需要予測の理解を深めていきましょう。なお、本書の冒頭でも述べましたが、需要予測モデルの実装やデータの細かい前処理などは、Excelやプログラミング言語をしっかり使っていく必要があり、本書の目的から少々離れてしまうので省略します。ここでは、大まかな処理や考え方の流れをつかみましょう。

➲ 商品IDごと 販売数ごとのPOSデータを使おう［図4-5-1］

商品IDごと、販売日ごとのPOSデータ

	A	B	C	D	E	F
1	item_id	item_category	calendar_date	day_of_week	order_quantity	
2	id_8280607	Vegetable	2021/1/25	Monday	4	
3	id_8280607	Vegetable	2021/1/26	Tuesday	11	
4	id_8280607	Vegetable	2021/1/27	Wednesday	7	
5	id_8280607	Vegetable	2021/1/28	Thursday	13	
6	id_8280607	Vegetable	2021/1/29	Friday	7	
7	id_8280607	Vegetable	2021/1/30	Saturday	8	
8	id_8280607	Vegetable	2021/1/31	Sunday	14	
9	id_8280607	Vegetable	2021/2/1	Monday	4	
10	id_8280607	Vegetable	2021/2/2	Tuesday	6	
11	id_8280607	Vegetable	2021/2/3	Wednesday	6	
12	id_8280607	Vegetable	2021/2/4	Thursday	14	
13	id_8280607	Vegetable	2021/2/5	Friday	4	
14	id_8280607	Vegetable	2021/2/6	Saturday	14	

・item_id：
商品ID
・item_category：
商品カテゴリ
・calendar_date：
販売日
・day_of_week：
販売日の曜日
・order_quantity：
販売数

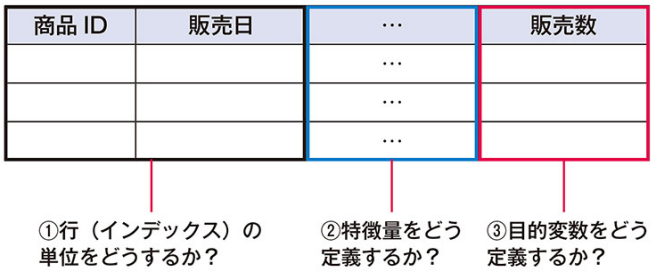
今回は、商品ごと・販売日ごとのPOSデータを使用しましょう。本章のデータはすべて「chap04_demand_forecast」フォルダにあります。生データは「dataset.csv」をご覧ください※11。

（※11）本書で使用するすべてのデータは、特に注釈がない限り、私がさまざまなオープンデータや実務経験を参考にしながら作成したダミーデータです。したがって、実際のビジネスやデータの細かい傾向や特徴とは乖離している部分があるかもしれません。

モデルへインプットするデータ構造を考える

さて、このデータは POS システムからそのまま抽出した生データという想定です。しかしこのままでは教師あり学習モデルを構築することはできません。モデルにインプットするデータにするために決めるべき点は 3 つあります。

➲ モデルへインプットするデータ構造を考える［図4-5-2］



1 つはデータの行（インデックス）単位です。今回は、「商品別・日別の販売数」を学習・予測するので、**行単位は「商品 ID× 販売日」でユニーク（一意）である必要があります**。今回の POS データはすでにそうなっているので、特に憂慮する必要はありません 。

すると、商品 ID ×販売日に合わせる形で、特徴量や目的変数をどうするか？という観点を明確にする必要があります。まず**目的変数に関しては、「商品 ID ごと販売日ごとの販売数」**とすればよさそうです。では、特徴量はどうすればよいでしょうか？

練習用ファイル：chap04_demand_forecast / dataset_preprocessed.csv

実践 特徴量を生成する（Feature Engineering）

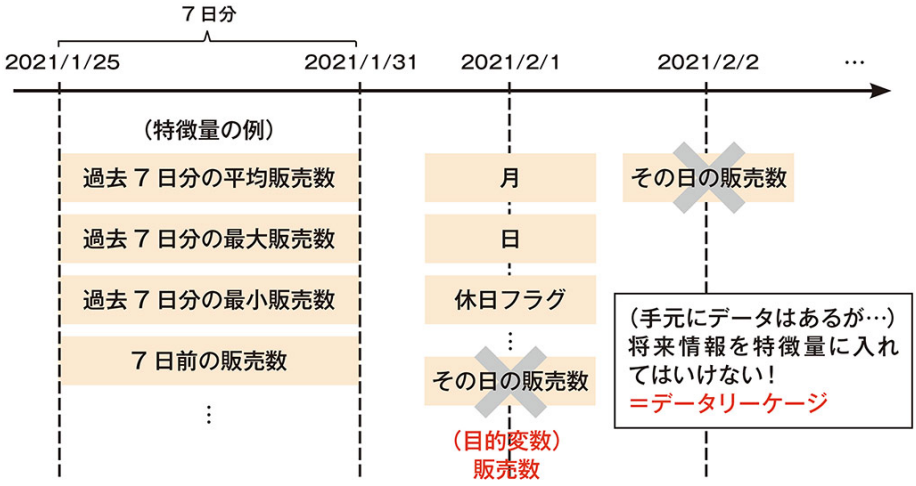
特徴量を定義する際は、目的変数である販売数に影響を及ぼすであろう要因を考える必要があります。そのためには次のような仮説を立てて、これを表現する特徴量を定義します。

➲ 特徴量を定義するための仮説の例［図 4-5-3］

・平日か休日かで販売数は変わりそうか？
・1 週間前の日の販売数に近しい販売数になるのではないだろうか？
・過去 1 週間の平均の販売数とも近しくなるのではないか？

また今回は、「**ある日において、明日の販売数を予測する**」という問題設定でした。したがって、ある日から考えて、その前日までにわかる情報から特徴量を考える必要があります。「その日の販売数」や「次の日の販売数」といった**その前日時点ではわからないデータを特徴量として含めないように注意する必要があります**。このような、**予測の段階ではわからない情報を特徴量として含めてしまうことを「データリーケージ」といい**ます。当たり前のように思うかもしれませんが、手元にある過去データからは、そういった情報がわかってしまいます。注意しないでいると、リーケージしているような特徴量を生成してしまうので、気をつけましょう。

➲ 適切な特徴量を定義・選択する［図4-5-4］



データリーケージに注意しつつ特徴量を生成（Feature Engineering）してみましょう。ある日の前日までにわかっている情報としては、たとえば次

のようなものが挙げられます。

➲ **ある日の前日までにわかっている情報［図 4-5-5］**

- **過去 7 日分の平均販売数**
- **過去 7 日分の最大販売数**
- **過去 7 日分の最小販売数**
- **7 日前の販売数**

これらの値から特徴量を生成できそうです。もちろん、過去 1 か月分の平均販売数といった特徴量も生成することができます。ただし、過去 7 日間の場合より多くのデータ期間を必要とするという点に注意が必要です。

また、月日や曜日といったカレンダー情報は、前日の段階でも当然ながらわかるので、そういった情報も特徴量に加えることができそうです。たとえば次のような情報です。

➲ **特徴量に加えられるカレンダー情報［図 4-5-6］**

- **月（月の値そのもの。1...12 の範囲となる）**
- **日（日の値そのもの。1...31 の範囲となる）**
- **休日フラグ（土日だったら 1、それ以外は 0）**
- **休日前フラグ（金土だったら 1、それ以外は 0）**

といった具合です。特に休日や休前日などは、飲食店であれば販売数は伸びそうですね※12。このように仮説を立てながら定義した特徴量を実際に生成します。その場合、生データを Excel や Python といったプログラミング言語で前処理する必要がありますが、本書ではすでに前処理したデータセットを用意しておきました。「dataset_preprocessed.csv」をご覧ください。定義としては、商品 ID ごと販売日ごとに以下のようになります。

（※ 12）本データでは、簡略化のため日本の祝日はないものとして作成しています。

➲ 特徴量の定義［図 4-5-7］

・order_quantity：販売数（目的変数に相当します）
・calendar_month：月
・calendar_day：日
・is_holiday：休日フラグ
・day_before_holiday：休前日フラグ
・1week_avg_order：過去 7 日分の平均販売数
・1week_min_order：過去 7 日分の最小販売数
・1week_max_order：過去 7 日分の最大販売数
・1week_ago_order：7 日前の販売数

もちろん、item_category（商品カテゴリ）といったカラムを用いることもできますが、今回はシンプルにモデリングしたいので、上記の特徴量を使って学習・予測をしていきましょう。

➲ 実際に特徴量を生成したデータセット［図 4-5-8］

（過去 1 週間から当日までの情報を用いた）

目的変数　特徴量

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	item_id	item_category	calendar_date	day_of_week	order_quantity	calendar_month	calendar_day	is_holiday	day_before_holiday	1week_avg_order	1week_min_order	1week_max_order	1week_ago_order
2	id_8280607	Vegetable	2021/2/1	Monday	4	2	1	0	0	9.142857143	4	14	4
3	id_8280607	Vegetable	2021/2/2	Tuesday	6	2	2	0	0	9.142857143	4	14	11
4	id_8280607	Vegetable	2021/2/3	Wednesday	6	2	3	0	0	8.428571429	4	14	7
5	id_8280607	Vegetable	2021/2/4	Thursday	14	2	4	0	0	8.285714286	4	14	13
6	id_8280607	Vegetable	2021/2/5	Friday	4	2	5	0	1	8.428571429	4	14	7
7	id_8280607	Vegetable	2021/2/6	Saturday	14	2	6	1	1	8	4	14	8
8	id_8280607	Vegetable	2021/2/7	Sunday	9	2	7	1	0	8.857142857	4	14	14
9	id_8280607	Vegetable	2021/2/8	Monday	7	2	8	0	0	8.142857143	4	14	4
10	id_8280607	Vegetable	2021/2/9	Tuesday	10	2	9	0	0	8.571428571	4	14	6
11	id_8280607	Vegetable	2021/2/10	Wednesday	9	2	10	0	0	9.142857143	4	14	6
12	id_8280607	Vegetable	2021/2/11	Thursday	9	2	11	0	0	9.571428571	4	14	14
13	id_8280607	Vegetable	2021/2/12	Friday	10	2	12	0	1	8.857142857	4	14	4
14	id_8280607	Vegetable	2021/2/13	Saturday	12	2	13	1	1	9.714285714	7	14	14
15	id_8280607	Vegetable	2021/2/14	Sunday	11	2	14	1	0	9.428571429	7	12	9
16	id_8280607	Vegetable	2021/2/15	Monday	7	2	15	0	0	9.714285714	7	12	7
17	id_8280607	Vegetable	2021/2/16	Tuesday	5	2	16	0	0	9.714285714	7	12	10
18	id_8280607	Vegetable	2021/2/17	Wednesday	3	2	17	0	0	9	5	12	9
19	id_8280607	Vegetable	2021/2/18	Thursday	8	2	18	0	0	8.142857143	3	12	9
20	id_8280607	Vegetable	2021/2/19	Friday	7	2	19	0	1	8	3	12	10
21	id_8280607	Vegetable	2021/2/20	Saturday	5	2	20	1	1	7.571428571	3	12	12
22	id_8280607	Vegetable	2021/2/21	Sunday	13	2	21	1	0	6.571428571	3	11	11
23	id_8280607	Vegetable	2021/2/22	Monday	7	2	22	0	0	6.857142857	3	13	7
24	id_8280607	Vegetable	2021/2/23	Tuesday	10	2	23	0	0	6.857142857	3	13	5
25	id_8280607	Vegetable	2021/2/24	Wednesday	5	2	24	0	0	7.571428571	3	13	3
26	id_8280607	Vegetable	2021/2/25	Thursday	9	2	25	0	0	7.857142857	5	13	8
27	id_8280607	Vegetable	2021/2/26	Friday	8	2	26	0	1	8	5	13	7

練習用ファイル：chap04_demand_forecast / train.csv、test.csv

実践 学習データとテストデータを決める

Section04 にて、手元のデータは、モデルの学習用データと、精度を評価するためのテストデータに分割する必要があると学びました。しかし、実際どのように分割すればよいのでしょうか？

何も考えないと、手元のデータの上半分を学習データ、下半分をテストデータなどと分割したくなりますが、そのようなやり方は基本的に NG です。そうすると、もし仮にそのデータが商品カテゴリ順などに並んでいたら、学習（もしくはテスト）データにしかないデータ、と偏りができてしまいます。基本的には**データをランダムに分割する方法が一般的**です。そうすることにより、学習データとテストデータの傾向が同程度となるので、学習データによるモデルの学習度合いを、適切にテストデータで評価できます。

しかしそれは、ある日だけのデータで学習・検証をする場合など、時系列データではない場合にのみ成り立ちます。なぜなら時系列データの場合、ランダムに分割してしまうと 2021 年 4 月のデータで学習して、2021 年 2 月のデータで予測するといった、**将来データで学習して過去データで予測するという、現実的にありえないことをしてしまうため**です。したがって時系列の場合は、過去 X 日分を学習データとして、直近 Y 日分をテストデータとする、といった分割方法になります。

今回は時系列データなので、後者の形となります。時系列でないデータの場合は次の第 5 章で解説します。

➲ 学習データとテストデータの分割方法は大きく分けて２通り［図4-5-9］

時系列データではない場合

商品 ID	販売日	…	販売数
AA	2021/2/1	学習データ	
BB	2021/2/1	テストデータ	
AA	⋮	⋮	⋮
OO	2021/2/1		30
PP	2021/2/1		17
⋮	⋮	学習データ	
ZZ	2021/2/1	テストデータ	

データをランダムに分割する

時系列データの場合

商品 ID	販売日	…	販売数
AA	2021/2/1	…	4
BB	2021/2/1	…	2
⋮	⋮	学習データ	
ZZ	2021/3/28	…	13
AA	2021/3/29	…	8
BB	2021/3/29	…	14
⋮	⋮	テストデータ	
ZZ	2021/4/4	…	24

データを時系列に分割する

分け方に明確な基準はありませんが、データ数が多いほうがよい結果を生むため、学習データを多くするのが一般的です。ただし、テストデータが少なすぎると、今度はテストデータでの精度評価の信頼性が低くなってしまいます。ランダム分割もそうですが、だいたい**学習データ：テストデータは、7:3、8:2、9:1 くらいで設定することが多い**です※13。今回は、ひとまず時系列の直近の１週間をテストデータ、それ以前を学習データとしましょう。Excel ファイルとしては、dataset_preprocessed.csv を分割し、それぞれ「train.csv」「test.csv」という形で格納しています※21。

➲ 学習データとテストデータに分割する［図4-5-10］

・学習データ（train.csv）
　：2/1 - 3/28 (8 週間、56 日) の期間のデータで学習（7,000 行）
・テストデータ（test.csv）
　：3/29 - 4/4 (1 週間、7 日) の期間のデータで予測（875 行）

（※ 13）とはいえ、明確な基準があるわけではないので、ケースバイケースで適宜決めていくことになります。

練習用ファイル：chap04_demand_forecast / test_result.xlsx

実践 教師あり学習（回帰問題）の予測結果を確認する

今回は、線形回帰モデルを 7,000 行の学習データで学習させ、テストデータ 875 行に対して予測し、その精度を確認してみましょう。本来、何かしら既存の業務において発注数と販売数という実績データがあれば、その予実差と教師あり学習モデルを比較するべきです。しかし今回はそのようなデータもなく、また実務的にも適切にデータを蓄積できていないケースがあります。そこで今回は Section02 にて、特徴量をある程度増やしていくことが有効であると説明したので、特徴量を一部使って学習する場合と今回生成した特徴量すべてを使う場合の両方で、予測結果を比較してみましょう。

➲ **予測結果の比較** ［図 4-5-11］

- **特徴量一部：過去販売数に関する以下の特徴量のみを使って学習したモデルで予測した場合**
  - ○「過去 7 日分の平均販売数」「過去 7 日分の最大販売数」「過去 7 日分の最小販売数」「7 日前の販売数」
- **特徴量全部：（上記に以下特徴量も加えて）今回生成したすべての特徴量を使って学習したモデルで予測した場合**
  - ○「月」「日」「休日フラグ」「休日前フラグ」

学習データで学習し、テストデータに対してモデルを適用した予測結果を、「test_result.xlsx」の「予測結果 _ 予測精度」シートの A ～ E 列に格納しています[※14]。上記 2 パターンごとの予測結果が、テストデータの商品別・日別で出力されているのがわかります。

（※ 14）実際は、すべてのパターンに関して、私が Python によるデータ処理や学習をして、その結果を Excel に出力している格好となります。

➲ テストデータにおける予測結果 [図 4-5-12]

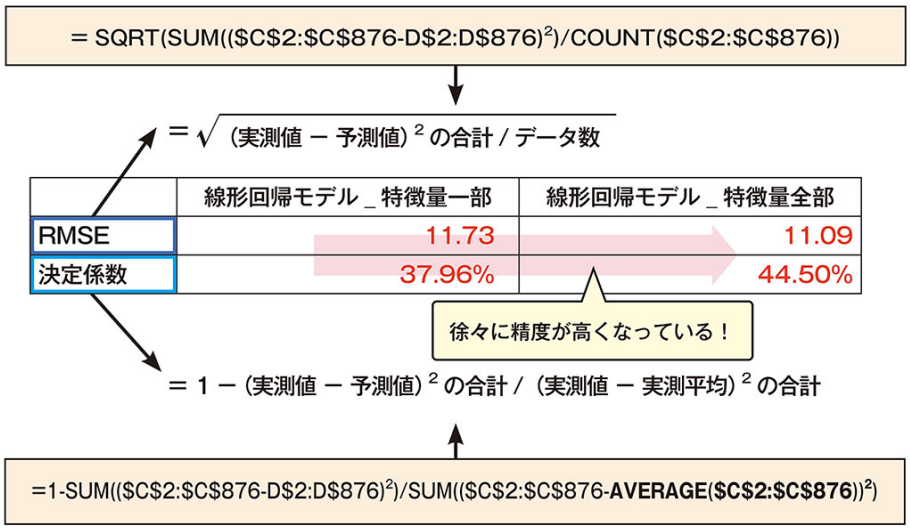
テスト期間における
商品別 日別の実測値

予測パターンごとの
商品別 日別の予測値

商品ID	日付	販売数	線形回帰モデル_特徴量一部	線形回帰モデル_特徴量全部
id_8280607	2021/3/29	8	9.844391331	6.02610433
id_8280607	2021/3/30	11	10.8931292	6.746414536
id_8280607	2021/3/31	10	10.22221843	6.325713231
id_8280607	2021/4/1	4	14.11490824	11.97167361
id_8280607	2021/4/2	10	10.70910417	13.48395109
id_8280607	2021/4/3	10	13.10425259	21.47295687
id_8280607	2021/4/4	11	8.923667149	14.05644607
id_5029823	2021/3/29	14	24.61070416	21.31291249
id_5029823	2021/3/30	9	27.22337083	22.73328946
id_5029823	2021/3/31	21	19.20163985	16.35500421
id_5029823	2021/4/1	28	23.84539486	22.68445488
id_5029823	2021/4/2	32	30.13323309	31.52320948
id_5029823	2021/4/3	34	23.6232876	33.1788071
id_5029823	2021/4/4	37	37.62094003	38.91600382
id_5644148	2021/3/29	24	21.68037973	19.12229091
id_5644148	2021/3/30	25	23.67036288	20.77802461

さて、予測値と実測値がただ出力されているだけだと、どうも解釈できません。そこで、まずは予測精度を確認してみましょう。今回は前述したRMSEと決定係数という精度評価指標で検証してみます。Excelでの計算は少々煩雑なので、私が計算した結果を、セルH4～I5に記載してあります。

➲ Excelによる RMSE と決定係数の計算 [図 4-5-13]



Section04 で紹介した定義に従って計算した結果を見てみましょう。RMSE と決定係数の要点は次の通りでした。

➲ **精度評価指標で押さえておくべきポイント［図 4-5-14］**

・RMSE は 0 に近づく（値が小さくなる）ほど、精度が高い
・決定係数は 100% に近づく（値が大きくなる）ほど、精度が高い

今回の結果を見てみると、特徴量を一部とした線形回帰モデルから、すべての特徴量を加えた線形回帰モデルへ精度評価指標が改善しているのが見てとれます。実務的には、さらに精度を高めていくために次のように改善していきます。

➲ **モデル精度向上のために考えられる改善策［図 4-5-15］**

・販売数に影響を与えそうな、そのほかの特徴量を増やしてみる
・モデルを線形回帰モデル以外の、より複雑なモデルも試してみる

今回は紙幅の関係上、これ以上モデリング自体は深く堀り下げませんが、一方で予測結果をもう少し考察してみましょう。

練習用ファイル：chap04_demand_forecast / test_result.xlsx

実践 予測結果を考察する

用意してある「予測結果 _ 可視化例」シートで分析をしましょう。たとえば今回の予測期間は 7 日分でしたが、日別で考えた際、「実際の販売数と予測値はどのような乖離になっていそうか？」を知りたいとします。その場合は、実測値と予測値を集計・可視化してみましょう。

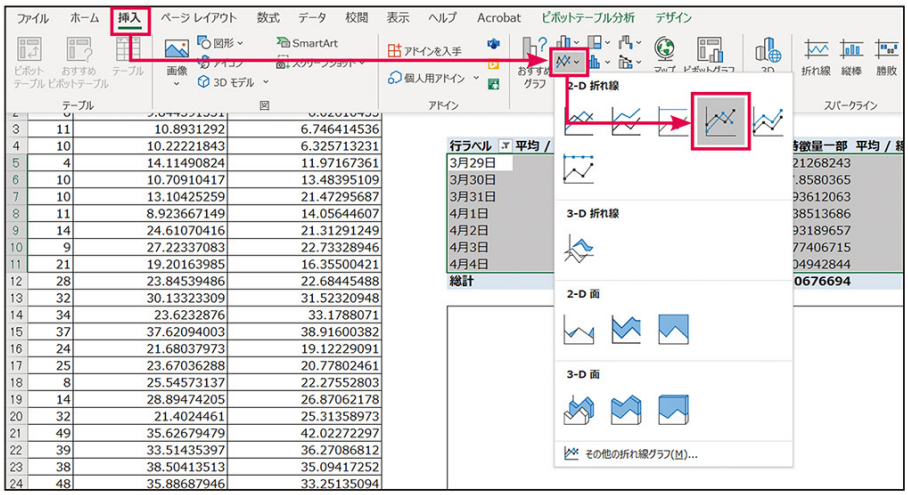
今回は Excel のピボットテーブルと折れ線を使ってみます※15。シートのセル G4 ～ J12 に、日別における以下の項目を、あらかじめピボットテーブルで作成してあります。

（※ 15）本書は Excel の使い方そのものに主眼を置いているわけではないので、細かい部分の説明は省略します。興味のある方は調べてみてください。

・実測値としての販売数の平均値
・特徴量を一部使用したモデルの予測値の平均値
・特徴量を全て使用したモデルの予測値の平均値

数値だけだと少々見にくいので、このピボットテーブルによる集計結果を折れ線グラフで可視化しましょう。セル G5 からセル J11 を選択して、[挿入] → [折れ線 / 面グラフの挿入] → [2-D 折れ線] → [マーカー付き折れ線] を選択しましょう（[図 4-5-16]）。すると、販売数・線形回帰モデル _ 特徴量一部・線形回帰モデル _ 特徴量全部の、日ごとの平均販売数（もしくは予測値）が折れ線グラフとして描画されているはずです（[図 4-5-17]）。

➲ ピボットテーブルによる集計結果を折れ線グラフで可視化する［図 4-5-16］

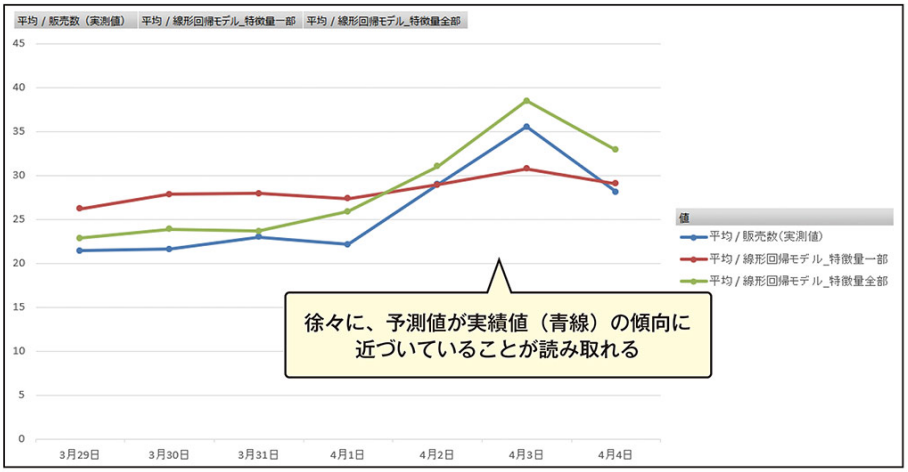


そのままのグラフでもよいですが、グラフ範囲を大きくしたり、凡例を上側に持ってきたり、字体を「Meiryo UI」に変更したりして体裁を整えた結果を [図 4-5-17] に示しています。青線が実際の販売数の日ごとの平均となっており、この線に近づくほど、予測結果が実態に即しているといえます※16。

特徴量を一部利用した線形回帰モデルの予測値である赤線は、少しだけ実測値の青線に近づいています。さらに特徴量をすべて使った緑線は、より実測値の傾向を表していると読みとれるでしょう。

（※ 16）Excel の解答例は「test_result_answer.xlsx」ファイルとして格納しています。

➲ 折れ線グラフの傾向を読み取る［図 4-5-17］



実測値を見ると、4/2（金）、4/3（土）、4/4（日）と販売数が伸びている、特に 4/3（土）が伸びている傾向となっており、一般的な飲食店の傾向を表していそうです。今回行った、過去販売数に関する特徴量だけを使った「線形回帰モデル _ 特徴量一部」に加えて、休日フラグ・休日前フラグといった特徴量を加えた「線形回帰モデル _ 特徴量全部」による予測は、曜日の傾向をつかみ、現実の事象をより正確に表現できている、と考えることができるでしょう。

練習用ファイル：chap04_demand_forecast / test_result.xlsx

実践 ビジネス上のKPIをシミュレーションする

最後に、モデルの精度評価をビジネス上の KPI で評価してみましょう。ビジネス上の KPI とは、それこそ売上やコストに直接つながるような指標を指します。ビジネス現場での活用を考えると、

モデルの精度評価→ビジネスインパクトの試算

という流れで評価することが一般的であり、できる限り後者の評価もしたいところです。今回のケースにおけるビジネス上の KPI は何が当てはまり

そうでしょうか？

１つ目は、発注数を試算する際に、機械学習により自動的に出力された予測値を参考情報とすることで、**発注数決定の時間を短縮**できそうです。稼働時間が節約でき、「人件費」の削減につながるでしょう。

２つ目は、本章の冒頭でも述べたように、予測による発注数と実際の販売数の予実差が縮まることで、「**廃棄や機会損失といったコストや売上損失**」を抑えられます。

前者の人件費に関しては、機械学習の予測の精度が直接効きにくい（精度が悪くとも自動化されていれば稼働時間は短縮されてしまうし、精度がよいとどの程度発注の決定時間が短縮されたかの計測が難しい）ため、今回は後者の廃棄や機会損失の改善度合いを考えてみます。もちろん実際に構築したモデルを実業務に適用し、過去の廃棄数や機会損失数との比較ができればベストですが、モデル構築時にすぐ業務適用はできないので、まずは何かしらの形でシミュレーションする必要があります。

前提として、このような廃棄数や機会損失数の正確な試算は非常に難しいという点を意識しておきましょう。その理由は次のようなものです。

➲ **正確な試算が困難な理由**［図 4-5-18］

- **商品によって廃棄までの日数（賞味期限・消費期限）が異なるため、ある日単体での予実差だけでは廃棄数は決められない**
- **過去の販売数は、もしかしたらすでに機会損失が生じたうえでの販売数である可能性があるため、過去データでの販売数との予実差では正確な機会損失数はわからない**

したがって実務では、上記のような**ビジネスの実態をできるだけ取り入れ、予測値と実測値、加えて賞味期限などのさまざまな情報をもとに、シミュレーションによる試算をする必要**があります。

とはいえあまり複雑な前提をおいて複雑な試算をすることは（もちろんよりリアルなケーススタディにはなりますが）難しく、本書の目的とは逸れてしまいます。そのため今回はモデルによる予測値から、できるだけシンプル

に試算してみます。かなり強い仮定となってしまいますが以下のような仮定をおいて試算しましょう。

➲ 今回のシミュレーションにおける仮定［図 4-5-19］

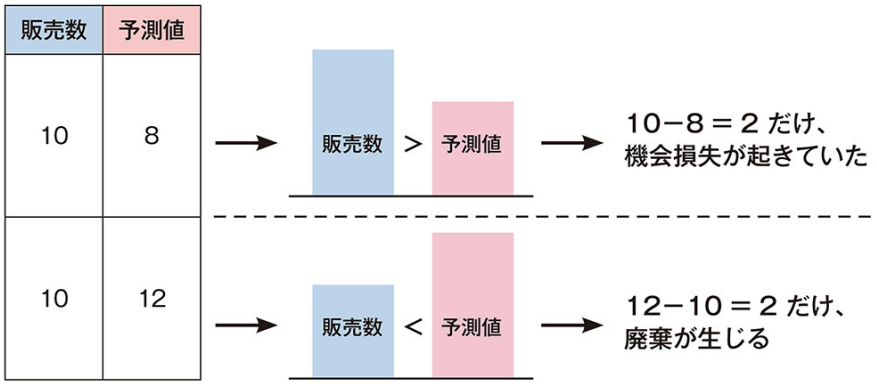
- 過去データの販売数から機会損失が起こっていたかどうかは、今回は考慮しない
- すべての商品商材は 1 日で廃棄されるものとし、前日までの在庫数は次の日に依存しないため、各日の予実差をそのまま廃棄する

つまり、各日において、下記のシミュレーションが成り立ちます。

➲ 各日に成立しうるシミュレーション［図 4-5-20］

- 「販売数 ＞ 予測数」であれば、販売数 - 予測数分機会損失が起きる
- 「販売数 ＜ 予測数」であれば、予測数 - 販売数分廃棄が生じる

➲ 予測の精度が、機会損失や廃棄の精度に繋がる［図 4-5-21］

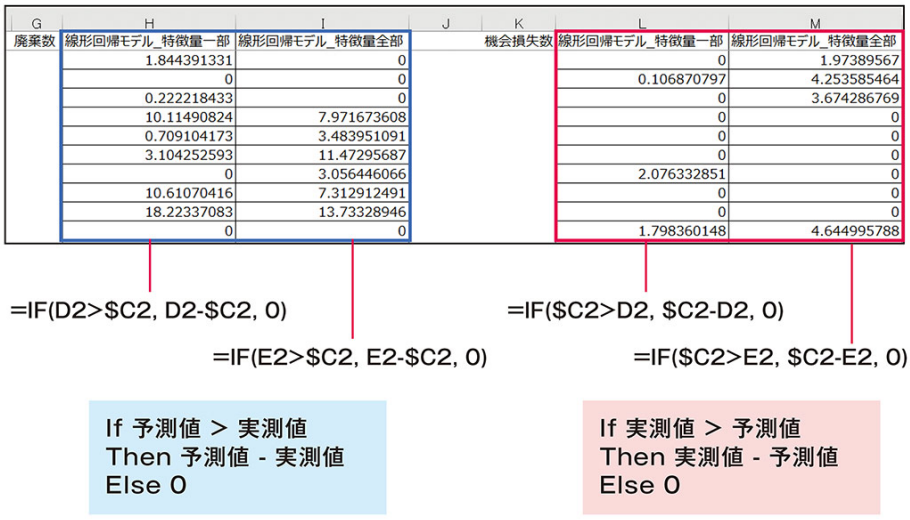


「予測結果 _ シミュレーション」シートを開いてください。今回の予測結果に関して、G ～ I 列で廃棄数、K ～ M 列で機会損失数を計算します。条件分岐（IF-ELSE）関数を使用して下記処理をすべての行に対して行っています。

➲ 予測値から廃棄数・機会損失数を定義する［図 4-5-22］

・廃棄数の場合、仮に予測値 > 実測値となっている行は、
「予測値 - 実測値」を廃棄数とする
・機会損失数の場合、仮に実測値 > 予測値となっている行は、
「実測値 - 予測値」を機会損失数とする

➲ 廃棄数と機会損失数を計算する［図 4-5-23］



最終的に、セル O3 ～ R7 にて、線形回帰モデル _ 特徴量一部・線形回帰モデル _ 特徴量全部に対して、廃棄数と機会損失数それぞれの合計や平均を、SUM 関数と AVERAGE 関数を用いて計算してみましょう。結果、RMSE や決定係数と同様に、モデルの精度がよくなるほど、廃棄数や機会損失数も改善していることが見てとれます。

➲ **モデルの精度に応じて、廃棄数や機会損失数の合計値や平均値は改善する**
［図4-5-24］

=SUM(H2:H876)　=SUM(I2:I876)
=AVERAGE(H2:H876)　AVERAGE(I2:I876)

		線形回帰モデル_特徴量一部	線形回帰モデル_特徴量全部
廃棄数	合計	5003.91	4798.20
	平均	5.72	5.48
機会損失数	合計	2842.487	2565.265
	平均	3.249	2.932

=SUM(L2:L876)　=SUM(M2:M876)
=AVERAGE(L2:L876)　=AVERAGE(M2:M876)

モデルの精度がよくなるほど、
廃棄数や機会損失数も減少させられている

より正確に試算するのであれば、商品ごとの売価と原価をもとに、商品ごとの「売価×機会損失数 + 原価×廃棄数」の全商品での合計などといった形で試算すると、より売上損失とコスト改善を金額ベースで把握できます。今回の試算はかなり簡易的でありましたが、諸条件を追加して複雑になったとしても「モデルの予測→ビジネス上の KPI をシミュレーションして試算する」部分は、本質的には同様の流れとなるので、常にビジネス的なインパクトを意識するようにしましょう。

今回は簡易的なシミュレーションに留めたけど、実務的には構築したモデルを実際のオペレーションに適用することで、ビジネス上の効果があるかどうかを検証する必要があるの。最初は実証実験といった形から入り、試行錯誤を繰り返すことで、徐々に運用にのせていく、という柔軟なやり方で実験を繰り返していくことが望ましいわね。

■ ここで学んだ重要トピック

- 教師あり学習、回帰問題
- 「学習」と「予測」
- 目的変数、特徴量、モデル、目的関数
- 線形回帰モデル（単回帰分析と重回帰分析）
- RMSE、決定係数

■ ステップアップにつながるトピック

- 決定木
- ランダムフォレスト（XGBooxt、LightGBM）
- ハイパーパラメータ
- 過学習／未学習
- 汎化性能
- クロスバリデーション
- グリッドサーチ、ランダムサーチ、ベイズ最適化

ステップアップにつながるトピックは、興味があったらさらに調べてみましょう。

巻末（261 ページ）からの情報も活用すれば、理解がより深まりそうですね。

自分で調べて身につけた知識は簡単には忘れないものなので、新しいことを学ぶときは自身で掘り下げて調べることがとても重要よ！

Chapter 5

ロジスティック回帰モデルで
ユーザーターゲティングを行う

Section

01 ユーザーターゲティングによりメール配信を高度化しよう

メール配信って、多くのお客さんに一括して同じ内容を流せるから便利な施策ですよね。

でも注意も必要よ。送ったメールが興味のない内容だと、受信拒否や配信解除されてしまう危険もあるでしょ。それでは顧客を失うも同然。だから、メール配信も狙いを定めてやらないと逆効果になる施策といえるわね。

え、そうなんですか？ 今度旅行企画会社に販促プランを提案する仕事があって、メールの一括配信の線で提案書を作ってました……。

配信する内容次第だから、必ずしもメールの一括送信がダメということじゃないけどね。

でも、どうやって配信相手を選べばいいんですか？ その人がどんな話題に興味あるかなんてわからないですよね。

これもデータサイエンスを使えば解ける問題ね。さぁ、提案書を直してその仕事ゲットしにいくわよ！

ここで学ぶこと

- ☑ 回帰問題と分類問題の違い
- ☑ ロジスティックス回帰モデルを実務で活用するための考え方
- ☑ 分類問題における精度評価指標

とある宿泊予約サイト運営会社の課題を考えてみよう

とある宿泊予約サイトの運営会社C社のケースを考えてみましょう[※1]。宿泊したいユーザーは、C社のサイトでユーザー登録を行い、宿泊場所や日程を検索し、予約します。ユーザーがこのサイトから宿泊予約を行うことで、C社は宿泊先から手数料を得るというビジネスモデルです。

C社のビジネス拡大にとって非常に重要なことは、ユーザー数を増やしつつ、1人ひとりのユーザーにC社のサイトを継続的に利用してもらうことです。ユーザーの体験を向上させるために、C社はさまざまな施策を打っていますが、そのうちの1つに、ユーザー登録をしている**ユーザーへのメール配信**があります。配信メールのコンテンツに予約サイトへのURLなどを埋め込んでおくことで、ユーザーがそのURLをクリックし予約のコンバージョン（成約、CV）が発生すれば、手数料としての売上が発生します。配信するメール内容は多岐に渡りますが、たとえば次のようなものがあるでしょう。

➲ **配信メールの内容例［図5-1-1］**

- **予約サイトの新規機能を幅広く認知させる**
- **毎年恒例の大型キャンペーン（宿泊●●％オフなど）の告知をする**

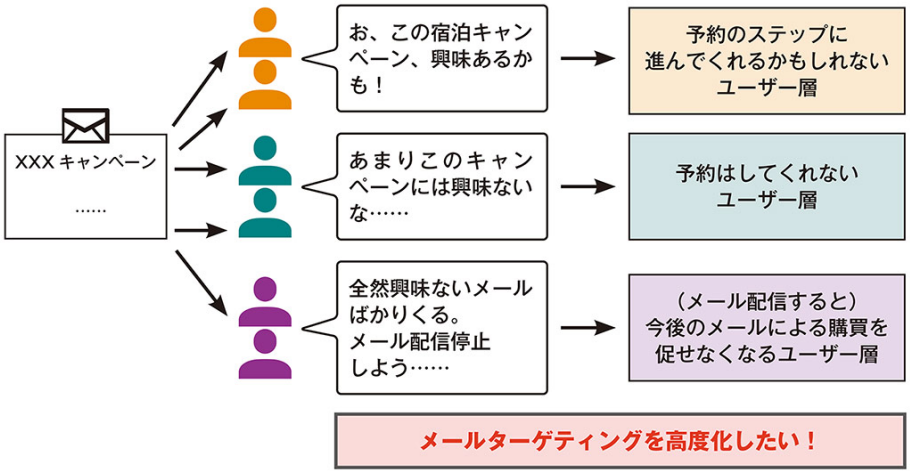
こういったコンテンツは、登録している全ユーザーに配信すべきメールだと考えられるので、誰に配信するかを気にする必要はあまりありません。

一方で今回は、比較的規模の小さく、かつ興味のあるユーザーが限られていそうな、季節限定のキャンペーン施策を打つことを検討しています。この場合は全ユーザーではなく、可能であればある程度**このメールに反応してくれそうなユーザーに絞って配信したいと**考えています。ターゲットとするユーザーが登録者全員ではないので、ユーザーによって、興味のある層・あまり興味のない層・まったく興味がなく、むしろメールを送るべきではない層、と分かれる可能性が高いと考えられます。仮に配信したいユーザーが（30代男性などと）確実に決まっていればよいですが、今回は明確に決まってい

（※1）宿泊予約サイトに限らず、ECサイトなど、同じような事業や課題をもっているようなケースでも、もちろんよいでしょう。

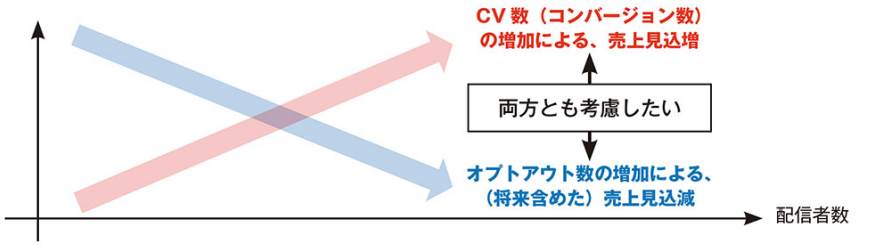
るわけではありません。そのため興味のありそうなユーザー群に配信することで、メール配信を高度化したいと考えています。

➲ 誰にこのメールを配信するべきか、を決めたい［図5-1-2］



このとき、「全員にメール配信をしてもいいのでは？」と考える人もいるでしょう。しかし、興味のないユーザーにメールを配信することはビジネス上のリスクにつながります。皆さんも思い当たるかもしれませんが、まったく興味のない内容が届いたと感じたユーザーは、メール配信を停止してしまう可能性があるためです。このような、**メール受信者が個別に受信拒否してしまう行為を「オプトアウト」**といいます。仮にユーザーがオプトアウトすると、そのユーザーにはほぼ半永久的にメールを送信できなくなってしまうので、将来的に発生するメール経由での期待売上（LTV※2）を損失してしまう可能性があります※3。

➲ CV 数とオプトアウト数の両方を考慮して配信者数を決められないか［図5-1-3］



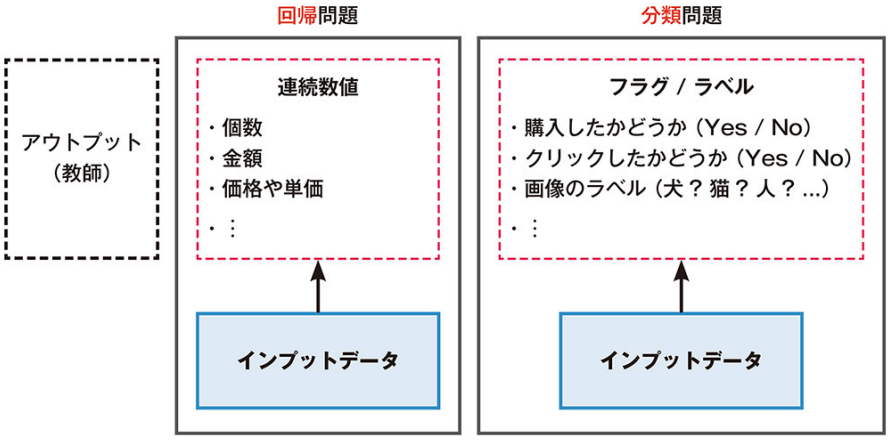
（※2）Life Time Value（顧客生涯価値）。ある1人のユーザーが将来全体を通して寄与するであろう価値のこと。
（※3）また導入しているメール配信システムやその運用業者によっては、1回配信するごとにいくらかかるといった配信コストがかさむケースもあります。このような場合は、より配信ユーザーを絞り込むインセンティブが働くでしょう。

そこで、ユーザーの属性情報や過去のサイト閲覧履歴などのデータを用いて、配信すべきユーザーとすべきではないユーザーを判別します。それによりメール配信によるCVとオプトアウトの両方を考慮しながら配信ユーザーを決定し（**ユーザーターゲティング**と呼ぶこともあります）、利益を最大化することを目指します。

データサイエンスで解くための問題設定

データサイエンスで解くための問題設定を詳細化しましょう。**「どのデータセットを用いて、目的変数をどう定義するか？」**という論点をクリアにします。大枠として、今回はユーザーごとに予約してくれるか／しないかを判別したいので、0／1の分類問題として解けそうです。各ユーザーが予約するか（1）／予約しないか（0）を学習することで、予約しそうなユーザーを特定する教師あり学習モデルを構築すればよさそうです。

➲ **回帰問題と分類問題。今回は分類問題となりそう**［図5-1-4］



その場合、「各ユーザーが予約したかどうか」を既存データから定義する必要があります。打ちたいキャンペーン施策経験が過去にあれば、そのデータから、実際にメールに反応して予約したユーザー／しなかったユーザーというデータを抽出すればよさそうです※4。ただし、もしこれまでキャンペーンをやったことがない場合は、たとえば似たような過去キャンペーンデータ

（※4）過去に同様のキャンペーンをやっていた場合、当時の配信ユーザー全員にそのまま再度配信すればよいのではないかと考える方もいるかもしれません。しかし、前回と同じ配信ユーザーに加えて追加でどのようなユーザーに配信すればよいか考えたい、過去の配信結果はよくなかったから再度どのようなユーザーに配信すればよいか考えたい、といった理由でモデル構築が必要な可能性が考えられるでしょう。

において、予約した／しなかったユーザーデータを利用する、といった方法も考えられます。今回は新たなキャンペーンを想定しているため、似たような過去キャンペーンデータを利用して、そのときの配信結果データを用いてモデルを構築しましょう。なお、その過去キャンペーンデータに関して、今回は「各ユーザーが予約したかどうか」を目的変数として定義しますが、必ずしも「予約したかどうか」である必要はありません。予約したユーザー数がかなり少ない場合は適切に学習できない可能性があるため、「予約というコンバージョン」より少し手前の段階を指標とするケースもあります。

➲ 予約より前の段階を指標にするケース［図5-1-5］

- 「予約申込ページをクリックしたかどうか」
- 「配信メールにある URL をクリックしたかどうか」

このように、「本来置きたいコンバージョンの手前」の行動指標（マイクロコンバージョン）を目的変数とするケースもあります。どこを目的変数とするかはケースバイケースですが、いずれにせよ、**どのようなデータで、どのようなユーザー行動を学習したいか？**を設計する必要があるでしょう。

➲ 目的変数をどう定義するか［図5-1-6］

過去データ

目的変数をどうするか？

ユーザー ID	年齢	性別	過去 PV 数	過去予約回数	…	予約したかどうか
id_111	20	1（男性）	3	1	…	0（No）
id_222	56	1（男性）	20	2	…	1（Yes）
⋮	⋮	⋮	⋮	⋮	⋮	⋮
id_888	35	0（女性）	1	0	…	0（No）
id_999	27	1（男性）	45	3	…	1（Yes）

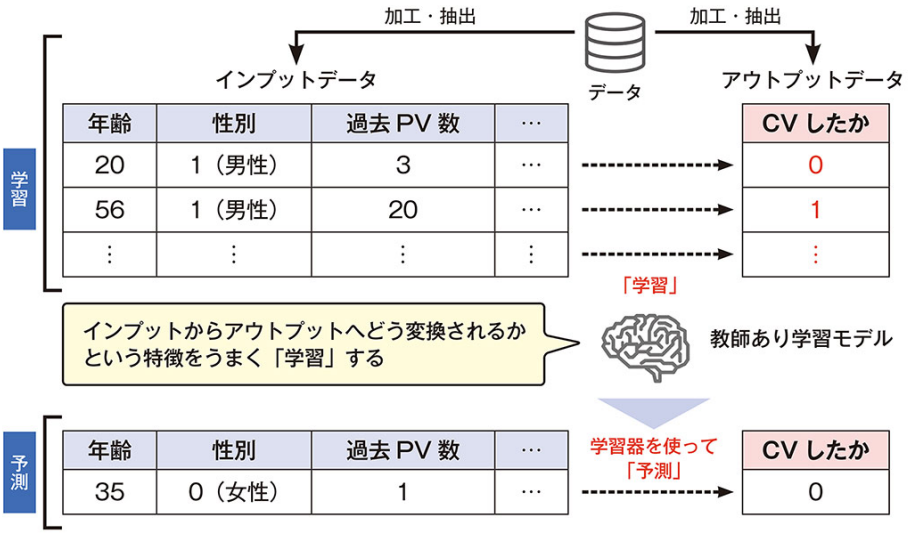
Section

02 分類問題の基本手法「ロジスティック回帰モデル」

教師あり学習（回帰問題）との共通点

分類問題といえども、基本的な考え方は回帰問題と変わりません。回帰問題でも分類問題でも、インプットとアウトプットを定義した学習データをもとに、「インプットからアウトプットへどう変換されるか？」という特徴を教師あり学習モデルが学習し、予測精度が向上していきます。

➲ 分類問題も、学習と予測のステップに切り分けることができる［図5-2-1］



回帰問題の場合、アウトプットデータである目的変数は「販売数」といった連続変数になります。一方で**分類問題の場合は、目的変数は XX したかどうか（0 か 1 か）といったカテゴリカル変数**です。正確には、今回のような0か1かといった2つの値を分類する問題の場合は**「二値分類」**と呼ばれます。また、たとえば「この画像は、猫か・犬か・橋か……人か」といった複数の値を分類する場合（マルチラベルといいます）は**「多値分類」**と呼ばれます。

多値分類に関しては、次の第6章で画像分類を取り扱う際に学びます。今回のケースでは、過去の類似キャンペーンデータをベースとして、次のようにデータを定義し、収集・加工します（具体的なデータのイメージは、後半の実践演習で詳しくみていきましょう）。

➲ 本ケースにおけるインプット・アウトプットデータ［図5-2-2］

・インプットデータ（特徴量）：ユーザーごとのさまざまな情報
例）年齢、性別、過去のPV[※5]数、過去の予約回数……など
・アウトプットデータ（目的変数）：ユーザーごとの予約したか（1）／していないか（0）

線形回帰モデルで解くことはできない？

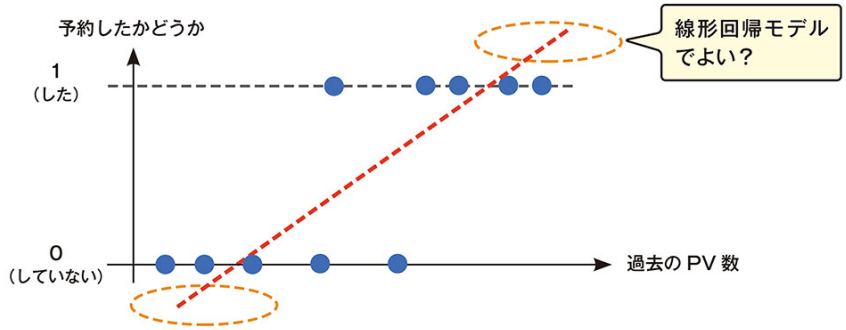
データが定義できれば、あとは学習するためのモデルを選択し、そのモデルを学習させます。このとき、たとえば第4章で学んだ線形回帰モデルを利用できるのでしょうか？

シンプルに考えるために、例としてユーザーごとの「過去のPV数」という1つの特徴量を用いて、各ユーザーが予約したかどうかという目的変数を学習させてみましょう。「Webサイトをより閲覧している＝PV数が大きいほど、宿泊やこのサービスに興味があることから予約しやすい」という傾向があるとすると、［図5-2-3］のようなデータとなりそうです。

これらのデータに対して、線形回帰モデルを適用しようとすると、当然右肩上がりの直線になるように学習するはずです（線形回帰モデルでは学習を繰り返すことで、最適な直線を求めるのでした。第4章参照）。しかし、今回の目的変数は0か1です。仮に直線として学習してしまうと、PV数がとても多い（少ない）ユーザーは1（0）を上回る（下回る）値として学習・予測してしまいそうです。したがって結論としては、**分類問題に対して線形回帰モデルを適用することは不適切であるといえます**。

（※5）ページビュー（PageView）。PV数は、ユーザーがWebサイトのページを閲覧した回数

➲ 0/1 の分類問題に対して、線形回帰モデルを適用してよいのか？［図5-2-3］

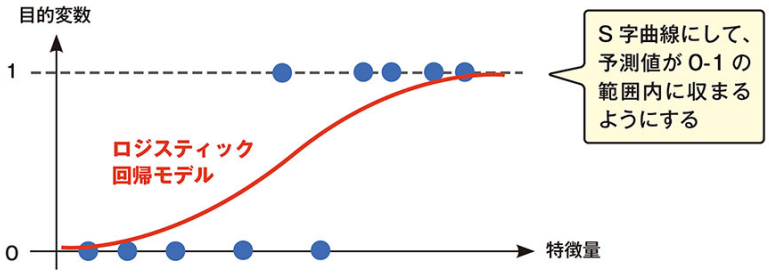


このように、**回帰問題と分類問題では、適用できるモデルの種類が異なることがある**という注意点を押さえておきましょう※6。

ロジスティック回帰モデルを導入する

そこで、線形回帰モデルを少し発展させた**「ロジスティック回帰モデル」**を適用させてみましょう。先ほどの課題は、目的変数の０／１に対して、直線では表現が難しいということでした。そこで直線ではなく、**０から１の範囲内に収まるＳ字曲線で表現**することで対応します。ロジスティック回帰モデルは、まさに［図5-2-4］にあるＳ字のような曲線になります。少し余談ですが、このロジスティック回帰モデルは線形回帰モデルをベースにして考えられています※7。

➲ S 字カーブにより、予測値を 0 から 1 の範囲内に収める［図5-2-4］

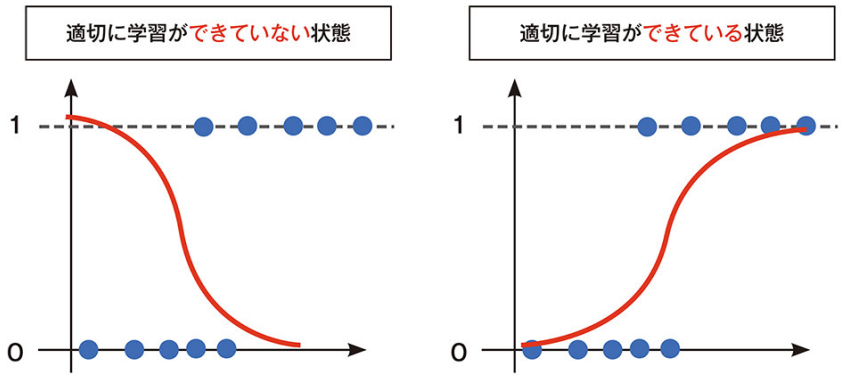


（※6）なお、前章で少し紹介した決定木、ランダムフォレスト、ニューラルネットワークといった回帰問題と分類問題の両方に対応しているモデルも存在します。
（※7）モデル名はロジスティック「回帰」モデルですが、あくまでロジスティック回帰モデルは分類問題のみに対応しています（回帰問題には使えません）。

ロジスティック回帰モデルで学習をする

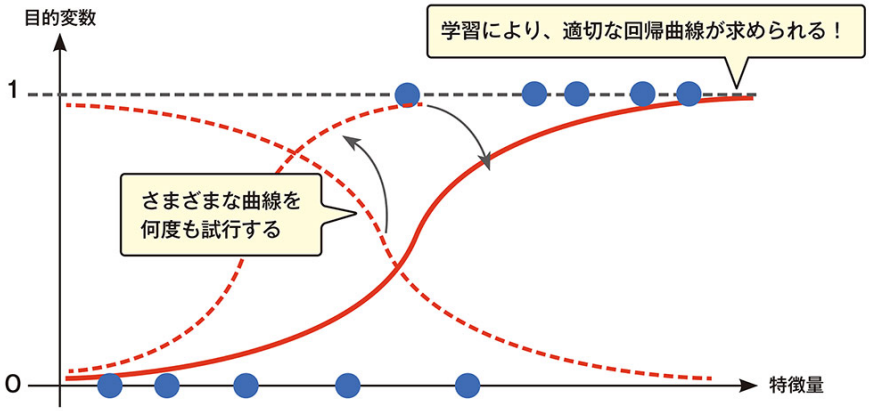
もう少し詳しくロジスティック回帰モデルの学習と予測の流れをみていきましょう。大まかな流れは線形回帰モデルと同様です。まず学習に関しては、**「特徴量と目的変数の関係性を最もよく表している曲線を求めること」**が必要となります。つまり、［図 5-2-5］の左図のような状態はロジスティック回帰モデルがデータに適切にフィットできていない＝学習できていない状態です。そうではなく、右図のように**モデルがデータに適切にフィットできている状態**が、学習できている状態であると捉えられます。もちろんデータの傾向はそうシンプルではないため、完全にフィットさせる＝すべてのデータが曲線上を通る、というのはほぼ不可能です。したがって、できるだけデータの傾向を表しているような曲線の形状になることが望ましい状態となります。

➲ 学習によりロジスティック回帰モデルをデータに適切にフィットさせる
［図 5-2-5］



モデルがデータにフィットしている状態とは、**データと曲線の差分が小さいこと**を指します。この差分のことを「目的関数」といいましたね（91 ページ参照）。ロジスティック回帰モデルにも同様に目的関数が存在しているので、**さまざまな曲線の形状を何回も試すことにより、目的関数が最小化するような回帰曲線を求める（＝学習する）**ことになります。

➲ 試行を繰り返すことで、適切な曲線を求める［図5-2-6］



Tips 交差エントロピー誤差関数

数学的な補足ですが、線形回帰モデルの場合は目的関数を「残差の二乗の合計（二乗誤差関数）」と定義し、その値を最小化しています。一方でロジスティック回帰モデルの場合は**「交差エントロピー誤差関数」**という目的関数を最小化しています。交差エントロピー誤差関数の紹介は、数学的な説明がかなり必要になってしまうので、本紙では説明を省きますが、先ほど説明したように、ロジスティック回帰モデルの回帰曲線とデータの差分を適切に数値化している関数だと押さえておけばよいでしょう。

ロジスティック回帰モデルで予測する

学習により適切なモデル（回帰曲線）を得ることができたら、予測のステップになります。学習済みモデルに対して、新たなデータを読み込ませます。たとえば特徴量として過去のPV数＝5のユーザーデータがあるとしましょう。そこから、ロジスティック回帰モデルにより、予測値を出力します。予測値は0から1の範囲内の値なので、これは**「予測確率」**となります。仮に予測値が0.3であれば、予約する（CVする）確率が30%である、と読み解けます。

実はこの予測確率をそのまま予測結果に利用することもできます。たとえば分類問題により営業先のアタックリストを作成した際に、「この顧客の予測成約確率は60%」と予測結果を得られたとしましょう。その場合、予測

成約確率の高い顧客から順にＮ人に営業をかける、といった使い方が想定できます。

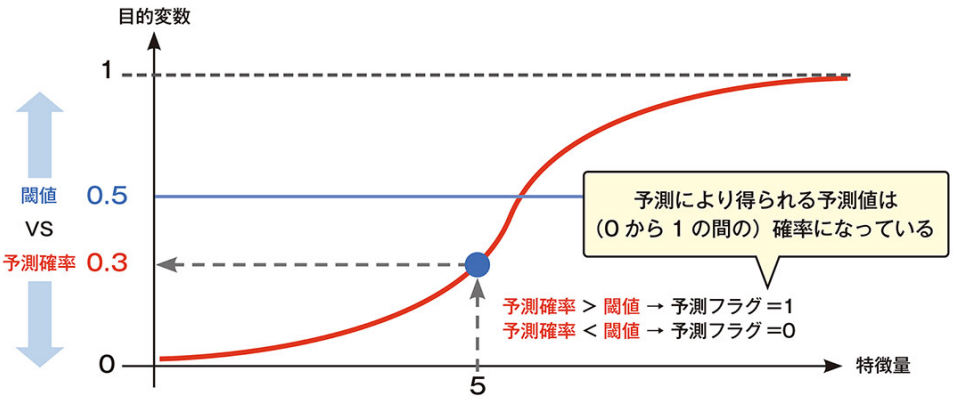
一方で、今回のケースでは予測値にもとづいて「メール配信するユーザー／しないユーザー」と分ける必要があるので、予測確率の状態だと誰に配信するのかが不明瞭です。そこで、**閾値を設けて、予測確率を 0/1 の予測フラグに変換**します。ロジックは以下のようにシンプルです。

➲ 閾値を設ける［図5-2-7］

・「予測確率 ＞ 閾値」であれば、予測フラグ＝1
・「予測確率 ＜ 閾値」であれば、予測フラグ＝0

仮に閾値を 0.5 とすると、予測確率が 0.3 であれば、予測フラグは 0 となり、そのユーザーは予約（CV）しないであろうと判断できます。

➲ 予測確率か予測フラグとして予測結果を出力できる［図5-2-8］



なお、この閾値は真ん中の 0.5（50％）である必要はなく、0.1 でも 0.9 でも構いません。どのように閾値を設定するかは難しいところですが、閾値を変えると予測フラグが変わるため、予測フラグと実測値の一致度合いを示す精度を見ながらチューニングすることが多いです。では、その精度の考え方を、次の Section でみていきましょう。

Section

03 分類問題における評価指標

Confusion Matrix（混同行列）

予測値を出せれば、その予測精度を評価する指標が計算できます。回帰問題の場合は、RMSE や決定係数といった指標を紹介しましたが、**分類問題は回帰問題と同様の指標を適用することはできません**。なぜなら、前述のとおり回帰問題の場合は予測値が連続数値であるのに対し、分類問題の場合は予測フラグ（もしくは予測確率）であり、数値の種類が異なるためです。そこで、分類問題の場合に使われる精度評価指標をいくつか紹介します。

精度評価指標を計算するために、まず**「Confusion Matrix」**（日本語では**「混同行列」**）と呼ばれる行列を作成する必要があります。名前は難しいですが、作り方は簡単です。まずは予測確率と設定した閾値から、予測フラグを算出します。そして実際の正解のフラグである実測値と突き合わせ、実測のフラグと予測のフラグの組み合わせの要素数を集計した表が Confusion Matrix と呼ばれます。

➲ 予測フラグと実測値から Confusion Matrix を作成できる［図5-3-1］

予測確率	予測フラグ	実測値
0.3	0	0
0.6	1	0
⋮	⋮	⋮
0.1	0	0
0.2	0	1
0.8	1	1
0.4	0	0

２×２の集計表を作る

（該当する行数をカウント）

		予測 1	予測 0
実測	1	400	100
	0	200	300

Confusion matrix（混同行列）

今回のような目的変数がYes/Noの二値分類の場合は、必然的に2×2の集計表になります。このConfusion Matrixはあくまで集計表なので、ここから、さまざまな指標を計算します。

Accuracy・Precision・Recall

Confusion Matrixから、精度評価指標を作成します。今回は、データサイエンスにおいてよく利用される3つの指標**「Accuracy・Precision・Recall」**を紹介します※8。Confusion Matrixがある前提で、3つの指標は以下のように定義されます。

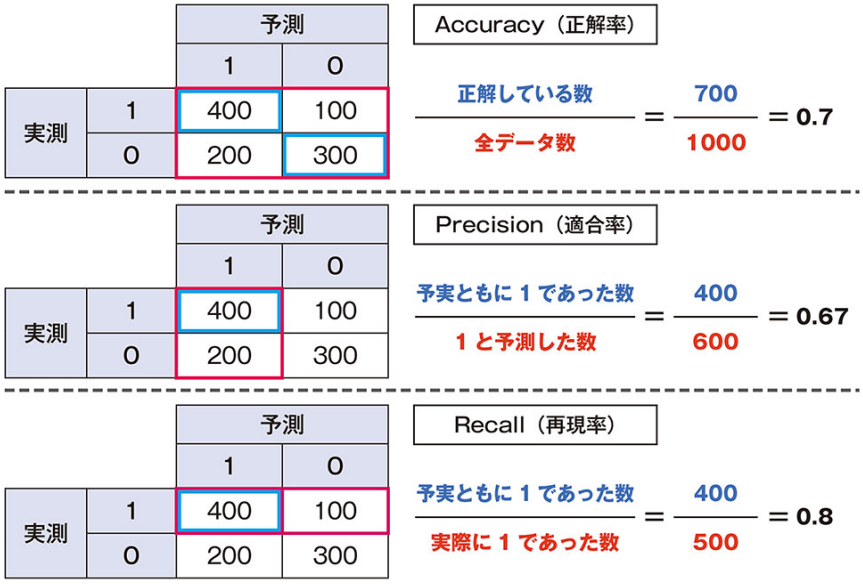
➲ 3つの指標［図5-3-2］

・Accuracy（正解率）
予測対象の全データ数のうち、0（1）と予測して実際に0（1）であった＝正解している数の割合を示す
・Precision（適合率）
1と予測したデータ数のうち、実際に1であった数の割合を示す
・Recall（再現率）
実際に1であったデータ数のうち、1と予測できた数の割合を示す

今回のケースで考えると、「1＝予約する、0＝予約しない」と定義できます。また、一般的には**「1を正例・0を負例」**と呼ぶことが多いです。

（※8）なお、当該3つの指標以外にも、真陰性率や偽陰性率といったさまざまな指標が存在します。実際医学系の業界などではしばしばこれらの指標も使われています。ただしこのようなすべての指標を紹介するのは本書の主旨から少し逸れてしまうので、あくまでデータサイエンスの業界で最低限押さえておくべき主要な指標のみ紹介します。

➲ Accuracy・Precision・Recall の定義と具体例［図5-3-3］



これらの指標は、感覚的に捉えれば、そこまで難しい定義ではありません。Accuracy はシンプルに予実が一致していた割合、Precision は予測した中で正解した割合、Recall は実際の 1 を 1 と予測できていた（取り込めていた）割合、と考えることができます。

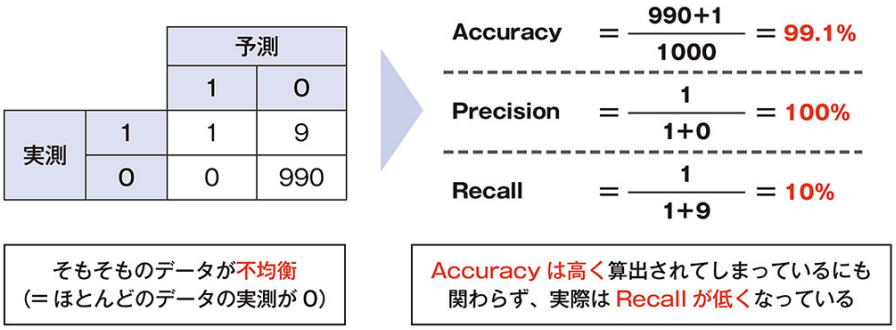
ここで、わざわざいくつもの指標を用意する必要があるのか？ わかりやすい Accuracy だけでよいのではないか？ と思った方もいるかと思います。そこで、Precision や Recall といった指標も観察する必要がある事象を考えてみましょう。

［図 5-3-4］のような Confusion Matrix となったとしましょう。この場合、予測データのほとんど（990/1000）が実際のフラグ =0 と偏っています。このようなデータを**「不均衡データ」**と呼びますが、不均衡データの場合の指標はどうなるでしょう。実は、Matrix における予測 = 0・実測 = 0 の要素にひっぱられてしまい、Accuracy が 99.1％と高くなっています。しかしよく見ると、Precision は 100％である一方で、Recall が 10％とかなり低い精度となっています。これは、大勢を占める実測フラグ = 0 にひっぱられ、モデルがほとんどのデータを 0 と予測してしまっていることが原因です。その

せいで、1と予測したデータがわずかしかなく、Precisionこそ高いものの、Recallが低くなってしまっているのです（1000データのうち999データが0なので、ほとんど0といっておけばよいようなモデルになってしまっています。これはあまり賢いモデルとは言い難いでしょう）。

このように、実は不均衡データだったために、Accuracyは高くなっていても、Recallが低くなっていて、実態として精度が高いモデルだとはいえないかもしれません。

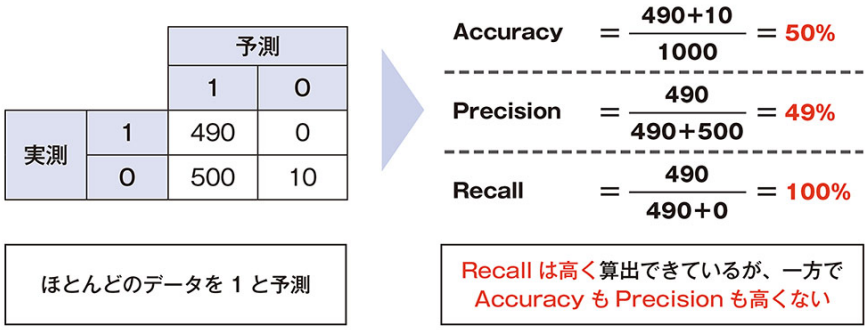
➲ 不均衡データの場合はAccuracyが高くなってしまう［図5-3-4］



ほかの例も考えてみましょう。［図5-3-5］のConfusion Matrixは、実測フラグ490のうち、すべてを1と予測しているので、Recallが100%になっています。しかし一方で、AccuracyやPrecisionが50%、49%と、Recallと比較するとあまり高くありません。よく見ると、1000データのうち990データを1と予測しているという予測モデルになっている（＝ほとんど1と予測しているだけ）ことが原因と考えられます。

このデータは不均衡データではないですが、ほとんどの予測データを1と予測してしまっているために、バランスのとれた予測モデルとはいえず、結果的にAccuracyやPrecisionが下がってしまっているといえるでしょう。

➲ Recall が高い一方で、Accuracy や Precision が高くないケースもある［図5-3-5］



このように、どれか1つの指標を見ているだけだと予測傾向の全体像を適切につかめなくなります。したがって、予測精度を評価する際には、**Confusion Matrix・Accuracy・Precision・Recall のすべてを満遍なく観察する必要**があります。また、Accuracy・Precision・Recall のどれをより重要視するかというのは、適用するビジネスシーンによっても変わってきます。たとえば、以下のような使い分けが想定されると考えられます。

➲ Accuracy・Precision・Recall の適用ケース［図5-3-6］

Accuracy が重要視される例

・不均衡データではない場合は一般的に重要視される

・たとえば、性別が判別しているユーザーデータを利用して、性別がわからないユーザーの性別（男性か女性か）を予測したい場合

Precision が重要視される例

・たとえば、スパムメールの判定（正例は「スパムである＝1」）でスパム＝1と判定されたら迷惑メールに選別されてしまう場合。仮に Precision が低いと、スパム確率がそこまで高くないメールも迷惑メールに選別されてしまい、スパムでないメールを見逃してしまう可能性があるため、Precision は高くあるべきと考えられる

Recall が重要視される例

・たとえば医学系などで、がん診断の判定（正例は「がんである＝ 1」）でがん＝ 1 と判定されたら精密検査をする場合。仮に Recall が低いと、本当はがんの患者ががんではないと診断されてしまう可能性があるため、Recall は高くあるべきと考えられる

このように、どの指標をより重要視するかというのは適用するビジネスシーンによって変わってくることを押さえておきましょう。

PrecisionとRecallを組み合わせた指標「F1スコア」

先の例のように、Precision か Recall のどちらかがより重要となってくるといった場合は、両方の指標を見比べていけばよいでしょう。しかし不均衡データなために Accuracy はあまり参考にならず、かつ Precision も Recall もどちらも重要といった場合は、**「F1 スコア」**を指標とすることもあります。F1 スコアは Precision と Recall を組み合わせた指標となっており、［図 5-3-7］のように定義されています。

➲ F1 スコアの定義式［図5-3-7］

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

・Precision（もしくは Recall）が 0 の場合

$$\text{F1 score} = \frac{2 \times 0 \times \text{Recall}}{0 + \text{Recall}} = 0$$

・Precision・Recall ともに 1 の場合

$$\text{F1 score} = \frac{2 \times 1 \times 1}{1 + 1} = 1$$

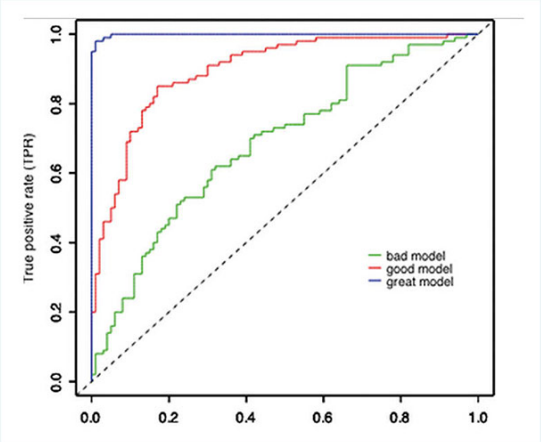
単純な平均ではないですが、Precision と Recall の平均を取ったような定義式となっています※9。特徴としては、Accuracy・Precision・Recall と同様に 0 から 1 の範囲の値となっていること、そして**Precision か Recall のどちらか片方でも 0 であれば、F1 スコアも 0 になってしまう**、という点です。つまり、Precision だけを重要視しすぎて Recall が低くなった場合は F1 スコアも低くなるように設定されているので、どちらもバランスよく高いときに、F1 スコアも高くなるようになっています。

Tips　予測確率を評価する指標 AUC

今回紹介した指標は、実はどれも「予測フラグ」を評価しています。よく考えると当たり前ですが、Confusion Matrix は予測と実測のフラグ同士から作られる表となっています。一方で、「予測確率」を予測結果として利用したい場合など、予測確率自体の精度評価をしたいときは、また異なる精度評価指標を使用する必要があります。予測確率を評価する指標には**「AUC」（Area Under the Curve）**が存在します。AUC も、Accuracy などと同様に、**1 に近づくほど精度が高い定義となっており、また 0.5 から 1 の間をとるような指標**となっています（正確には 0.5 未満の値をとる可能性もなくはないのですが……）。

しかし、AUC の導出過程は少々複雑なため、紙面の関係もあり今回は細かく紹介はしませんが、そのような指標があることは押さえておくとよいでしょう。

➲ AUC のイメージ図※10［図 5-3-8］



（※ 9）細かい話ですが、数学的には、一般的な平均のことを「算術平均」と呼びますが、F1 スコアの定義式は「調和平均」と呼ばれています。
（※ 10）出典：https://stats.stackexchange.com/questions/160935/relationship-between-roc-receiveroperating-characteristic-curve-and-cross-ove

Section

04 実践：宿泊予約サイトのユーザーデータを活用しよう

練習用ファイル：chap05_mail_targeting / dataset.csv

実践 データの確認

最後に、実践演習を通して分類問題のビジネス適用のイメージを深めていきましょう。冒頭で述べたように、**宿泊予約サイトのユーザーデータを活用して、メール配信の際にユーザーターゲティングを行い、配信の高度化を試みましょう**。今回は、各ユーザーの属性データや過去履歴データを使用します。属性データというのは年齢や性別といった情報を指し、よく「デモグラフィックデータ」といわれるものです。また過去履歴データは、ユーザーの過去の行動履歴に紐づく情報であり、今回使用するような過去の予約回数やPV数といったデータが考えられます。今回使用するデータには、以下のカラム情報が付与されています。「chap05_mail_targeting」フォルダ配下の「dataset.csv」に生データを格納してあります。

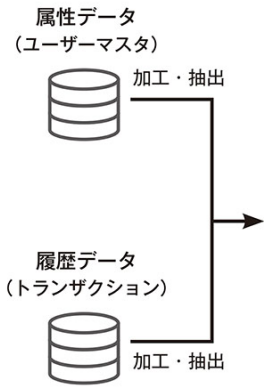
➲ 今回使用するデータのカラム情報［図5-4-1］

- user_id：ユーザーID
- is_booked：予約したかどうか（1：した・0：していない）
  ※今回の目的変数に相当します。
- age：年齢
- gender：性別（1：男性・0：女性）
- avg_pv_per_session：過去のセッションあたり平均PV数[※11]
- avg_travel_distance：過去の平均旅行距離
- past_cv_count：過去の予約回数

（※11）セッションとは、アクセスの開始から終了までの一連の行動を示しており、そのサイトにアクセスしてから、サイトから出て行くかブラウザを閉じるまでが1セッションとなります。したがって「セッションあたりPV数」とは、1セッションで何ページを閲覧したか、を指すことになります。

年齢や性別といった属性データ（ユーザーマスタということが多いです）と、過去の履歴データ（トランザクションということもあります）は別々になっていることが多く、今回はそれを user_id で紐づけています。実務的にはこういった元データから今回のようなデータに加工する**データの前処理**を行うことも多いですが、今回はその過程は省略します※12。

➲ ユーザーごとのデータを活用しよう［図5-4-2］



ユーザーごとのデータ

	A	B	C	D	E	F	G
1	user_id	is_booked	age	gender	avg_pv_per_session	avg_travel_distance	past_cv_count
2	id_1000027	0	45	1	6	4123.2	1
3	id_1000167	0	33	1	2	294.2	1
4	id_1000252	0	66	1	21	2084.1	3
5	id_1000321	1	39	1	16	776.3	4
6	id_1000393	0	32	1	53	934	1
7	id_1000546	1	34	1	14	1410.4	2
8	id_100056	0	42	1	10	1075.1	0
9	id_1000715	0	39	0	160	1528.6	3
10	id_1000750	1	28	1	44	943.8	6
11	id_1000786	1	54	0	131	1057.4	3
12	id_100082	1	54	1	3	965.8	3
13	id_1001000	0	50	1	47	1436	3
14	id_1001015	1	49	1	5	357.3	0
15	id_1001061	0	34	1	46	450.5	2

前章では Feature Engineering を考えましたが、今回はすでに user_id ごとにデータが整理されているので、元データのままモデルの学習をしましょう。つまり、ユーザーの年齢や性別、過去の PV 数や旅行距離や予約回数といった特徴量から、予約したかどうかを予測するモデルを構築することになります。

➲ 特徴量と目的変数［図5-4-3］

・特徴量

○ age、gender、avg_pv_per_session、avg_travel_distance、past_cv_count

・目的変数

○ is_booked

（※12）実務的にこういった細かいデータの前処理は、データベースへの問い合わせ言語 SQL や、Python などのプログラミング言語を使用してなされることが多いです。

練習用ファイル：chap05_mail_targeting / train.csv、test.csv

実践 全体の処理の流れを理解する

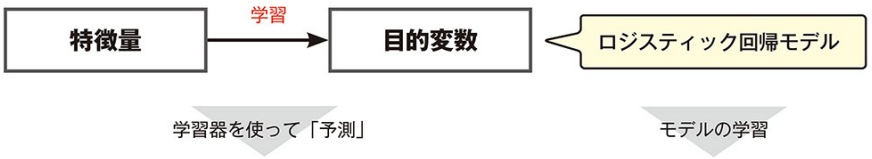
モデル構築の前に、一度全体の処理の流れを押さえておきましょう。基本的には回帰問題の場合と同様の「学習」と「予測」となります。そこで、dataset.csv を分割し、以下 2 つのデータセットとして格納します。

➲ 今回使用するデータセット［図5-4-4］

- **学習データ（train.csv）**
  モデルを学習するための「学習データ」
- **テストデータ（test.csv）**
  学習モデルの精度を評価するための「テストデータ」

➲ 全体の処理の流れ［図5-4-5］

①学習（train.csv：学習データ）



②予測（test.csv：テストデータ）

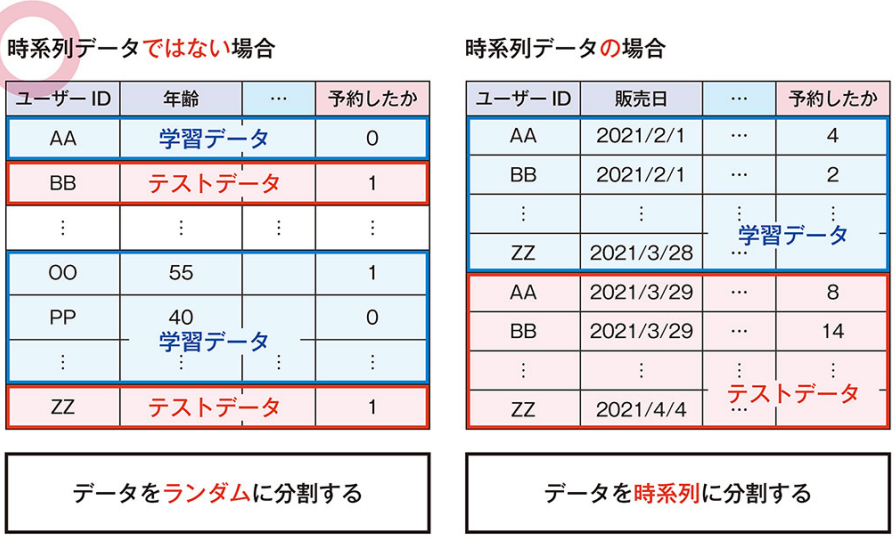
学習データでモデルの学習を行い、テストデータに対してモデルを適用し、精度評価を行う

最初に、**モデルを学習させるための学習データ**を用意します。今回は先ほど紹介したロジスティック回帰モデルを使用しましょう。次に、学習したモデルとテストデータの特徴量を用いて予測し、**実測値と比較して精度評価**をします。

なお、学習するためのデータと予測精度を検証するテストデータを分ける方法は主に 2 通りあると前述しました。回帰問題による需要予測モデル構築

の際は、時系列に沿ったデータの持ち方をしていたので、時系列でみて過去分と直近分で分割しました。一方で今回はあくまでユーザーごとのデータ、つまり時系列なデータではありません[※13]。このような場合は、**ランダムにデータを分割し、学習用とテスト用でデータを分ける**のが一般的です。したがって、今回の train.csv と test.csv は、元のデータをランダムに分割することで作成しています[※14]。

➲ **今回はデータをランダムに分割する［図5-4-6］**



別にランダムに分割せずとも、「データの上7割を学習用データ、下3割をテスト用データとすればよいのではないか？」と思った方もいるかもしれませんが、それは基本的には NG です。なぜなら、たとえばそのデータが年齢順やユーザー登録がされた順に並んでいたりなどしたら、分割した両者のデータ間で傾向がまったく異なってしまうためです。**できるだけ学習用のデータと精度テスト用のデータで同様の傾向とし、適切な精度評価をするために、元データをシャッフルし、ランダムに分割**するという方法が一般的です。

（※13）過去の PV 数や予約数といった特徴量もあることから、本質的には今回のようなデータも「時間的」な情報や概念は持っています。しかしあくまでデータの持ち方として、時間情報が行単位に含まれているかどうか（今回は行がユーザー単位なので含まれない）、といった観点から判断することが一般的です。
（※14）実際は Python などのスクリプト言語を用いて、ランダムにデータをシャッフルして分割することが多いです。

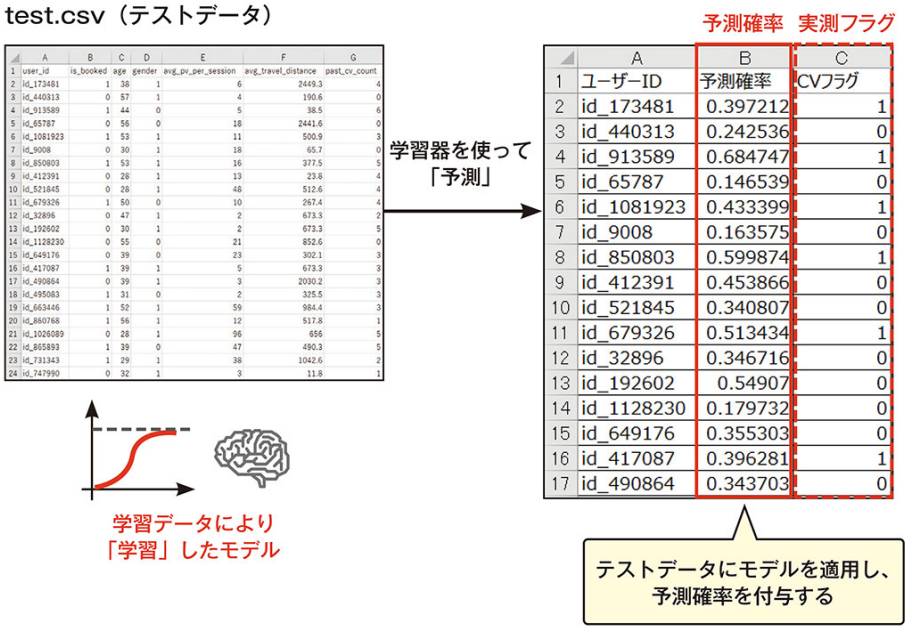
練習用ファイル：chap05_mail_targeting / test_result.xlsx

実践 教師あり学習（分類問題）の予測結果を確認する

今回は、学習（train）データ 11,258 行のデータ、すなわち 11,258 ユーザーのデータに対してロジスティック回帰モデルを適用・学習させた後、テスト（test）データ 4,825 行（ユーザー）に対して予測し、その精度を確認してみましょう。学習モデルをテストデータに対して適用した予測結果を、「test_result.xlsx」の「予測結果 _ 閾値別予測フラグ」シートの A ～ C 列に格納しています（実際は、Python によるデータ処理や学習をして、その結果を Excel に出力している格好となります）。

test_result.xlsx には、テストデータ（test.csv）の全 user_id それぞれの「予測確率 = CV してくれそうな確率」が付与されています。これは、学習データ = train.csv でロジスティック回帰モデルを学習させ、そのモデルをテスト用データに適用することで得られた値になります。その予測確率と、正解の値である実測フラグとを比較することで、精度を評価します。

➲ モデルを学習させ、テストデータにモデルを適用し、予測結果を得る［図5-4-7］



	A	B	C	D	E	F	G
1	user_id	is_booked	age	gender	avg_pv_per_session	avg_travel_distance	past_cv_count
2	id_173481	1	38	1	6	2449.3	4
3	id_440313	0	57	1	4	190.6	0
4	id_913589	1	44	0	5	38.5	6
5	id_65787	0	56	0	18	2441.6	0
6	id_1081923	1	53	1	11	500.9	3
7	id_9008	0	30	1	18	65.7	0
8	id_850803	1	53	1	16	377.5	5
9	id_412391	0	28	1	13	23.8	4
10	id_521845	0	28	1	48	512.6	4
11	id_679326	1	50	0	10	267.4	4
12	id_32896	0	47	1	2	673.3	2
13	id_192602	0	30	1	2	673.3	5
14	id_1128230	0	55	0	21	852.6	0
15	id_649176	0	39	0	23	302.1	3
16	id_417087	1	39	1	5	673.3	3
17	id_490864	0	39	1	3	2030.2	3
18	id_495083	1	31	0	2	325.5	3
19	id_663446	1	52	1	59	984.4	3
20	id_860768	1	56	1	12	517.8	1
21	id_1026089	0	28	1	96	656	5
22	id_865893	1	39	0	47	490.3	5
23	id_731343	1	29	1	38	1042.6	2
24	id_747990	0	32	1	3	11.8	1

	A	B	C
1	ユーザーID	予測確率	CVフラグ
2	id_173481	0.397212	1
3	id_440313	0.242536	0
4	id_913589	0.684747	1
5	id_65787	0.146539	0
6	id_1081923	0.433399	1
7	id_9008	0.163575	0
8	id_850803	0.599874	1
9	id_412391	0.453866	0
10	id_521845	0.340807	0
11	id_679326	0.513434	1
12	id_32896	0.346716	0
13	id_192602	0.54907	0
14	id_1128230	0.179732	0
15	id_649176	0.355303	0
16	id_417087	0.396281	1
17	id_490864	0.343703	0

練習用ファイル：chap05_mail_targeting / test_result.xlsx

実践 閾値ごとに予測フラグを計算する

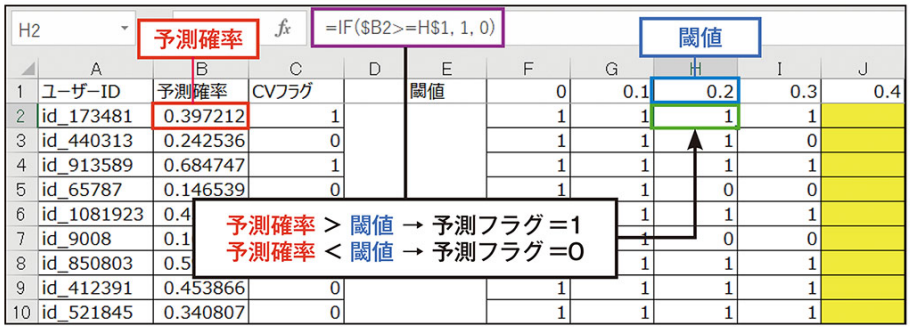
今回は、より実践的に考えてみましょう。先ほど、予測確率から閾値に応じて予測フラグを算出すると述べました。そこでこの演習では、出力された**予測確率から、閾値を 0 から 0.9 まで 0.1 ずつずらした際の予測フラグを算出**しましょう。「予測結果 _ 閾値別予測フラグ」シートを開いてください。B 列にユーザー ID ごとの予測確率があります。またセル F1 からセル O1 にかけて、閾値があります。この各閾値と各予測確率とを照らし合わせて、［図 5-4-8］のロジックを IF 関数で表現します。

➲ 予測値から予測フラグへ変換する［図 5-4-8］

・「予測確率 ＞ 閾値」であれば、予測フラグ＝1
・「予測確率 ＜ 閾値」であれば、予測フラグ＝0

閾値が 0.2 の場合の数式：=IF($B2>=H$1, 1, 0)

➲ 閾値に応じて、予測確率を予測フラグに変換する［図 5-4-9］



	A	B	C	D	E	F	G	H	I	J
1	ユーザーID	予測確率	CVフラグ		閾値	0	0.1	0.2	0.3	0.4
2	id_173481	0.397212	1			1	1	1	1	
3	id_440313	0.242536	0			1	1	1	0	
4	id_913589	0.684747	1			1	1	1	1	
5	id_65787	0.146539	0			1	1	0	0	
6	id_1081923	0.4					1	1	1	
7	id_9008	0.1					1	0	0	
8	id_850803	0.5					1	1	1	
9	id_412391	0.453866	0			1	1	1	1	
10	id_521845	0.340807	0			1	1	1	1	

「$B2」が予測確率、「H$1」が閾値に相当します。これを全予測確率、全閾値で計算できます。試しに、閾値が 0.4、0.5 の部分を空欄しておいたので計算してみてください[※15]。当然ですが、閾値が上がるほど、予測フラグも 1 が少なく、0 が多くなってくるはずです。

（※ 15） Excel の解答例は「test_result_answer.xlsx」ファイルとして格納しています。

練習用ファイル：chap05_mail_targeting / test_result.xlsx

実践 閾値ごとにConfusion Matrixを計算する

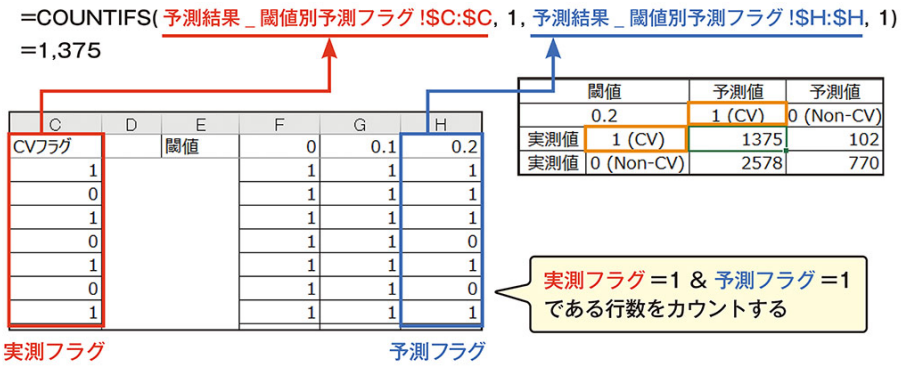
閾値ごとの予測フラグが計算できたら、次いで、**実測フラグと照らし合わせて、閾値ごとの Confusion Matrix を計算**します。「予測結果 _ 予測精度」シートの B 列から E 列に、Confusion Matrix が作成されています。ここではシンプルに、閾値ごとの予測フラグと実測フラグの値の組み合わせが (0, 0), (0, 1), (1, 0),(1, 1) となっているデータ数をそれぞれカウントするだけです。Excel の詳細には踏み込みませんが、この場合は **COUNTIFS 関数**が使えそうです[※16]。たとえば閾値が 0.2 の場合は以下のようになります。

➲ COUNTIFS 関数を使った数式例［図 5-4-10］

```
=COUNTIFS( 予測結果 _ 閾値別予測フラグ !$C:$C, 1, 予測結果 _ 閾値
  別予測フラグ !$H:$H, 1)
```

「予測結果 _ 閾値別予測フラグ !$C:$C」が実測フラグに相当し、ここは閾値によらず固定です。また「予測結果 _ 閾値別予測フラグ !$H:$H」が予測フラグに相当し、閾値によって対象とする列が変わります。第 2、第 4 引数は、対象とする組み合わせの値で、［図 5-4-11］のオレンジ部分 (1,1) に該当する

➲ 実測フラグと、閾値に応じた予測フラグから Confusion Matrix に変換する［図 5-4-11］



(※ 16) COUNTIF 関数は、条件に当てはまるセルの数をカウントする関数のため、今回の Confusion Matrix の集計に適しています。

データ数を計算する場合は、それぞれの引数を 1,1 とします。

これを 4 象限の組み合わせそれぞれで計算することで Confusion Matrix を作成できます。ここも同様に、閾値が 0.4、0.5 の部分を空欄にしておいたので、Confusion Matrix を作成してみてください。これで、閾値が 0 から 0.9 まで 10 通りの Confusion Matrix が作成できました。

練習用ファイル：chap05_mail_targeting / test_result.xlsx

実践 閾値ごとのモデル精度評価指標を計算する

閾値ごとに Confusion Matrix が作成できたら、いよいよ精度評価が計算できそうです。Confusion Matrix から Accuracy、Precision、Recall を計算する際の定義は、すでに紹介しているので、あとは愚直に閾値ごとの Confusion Matrix から計算するだけです。閾値が 0.2 の場合の Accuracy の計算は以下のようになります。Excel での計算の場合は、「=(D16+E17)/SUM(D16:E17)」などとセルを指定して計算してもよいでしょう。

Confusion Matrix から精度評価指標を計算する［図5-4-12］

Confusion Matrix から
精度評価指標を計算する

閾値	モデル精度評価指標			
	Accuracy(正解率)	Precision(適合率)	Recall(再現率)	F1
0	30.6%	30.6%	100.0%	46.9%
0.1	32.2%	31.0%	99.5%	47.3%
0.2	44.5%	34.8%	93.1%	50.6%
0.3	62.8%	43.4%	71.2%	53.9%
0.4				
0.5				
0.6	71.0%	69.9%	9.3%	16.4%
0.7	70.0%	72.6%	3.0%	5.8%
0.8	69.6%	100.0%	0.7%	1.3%
0.9	69.4%	0.0%	0.0%	0.0%

閾値		予測値	予測値
0.2		1 (CV)	0 (Non-CV)
実測値	1 (CV)	1375	102
実測値	0 (Non-CV)	2578	770

Accuracy（正解率）

$$\frac{1375+770}{1375+102+2578+770} = 44.5\%$$

・Accuracy の場合、(1375+770) / (1375+102+2578+770) = 44.5%
・Precision の場合、1375 / (1375+2578) = 34.8%
・Recall の場合、1375 / (1375+102) = 93.1%

また Precision と Recall が計算できたら、次のように F1 スコアを計算できます。

➲ F1 スコアの算出［図5-4-13］

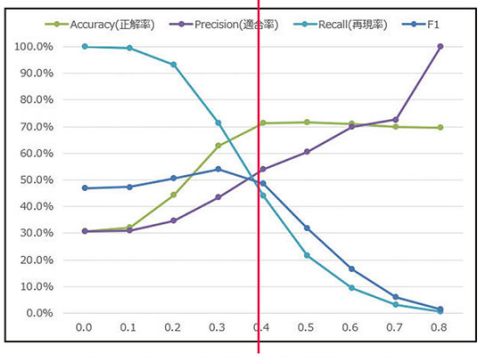
F1 スコア ＝ 2 * Precision * Recall / (Precision + Recall)

ここでも閾値が 0.4、0.5 の部分を空欄にしておいたので、それぞれの精度評価指標を計算してみてください。

また、全閾値の精度評価指標が計算できたら、グラフ化して少し解釈をしてみましょう。表下セル G16 あたりに「横軸が閾値・縦軸が各精度評価指標」をとったグラフを描画しています。全閾値の精度評価指標が出揃ったら［図 5-4-14］のようになるでしょう。この結果を見ると、4 つの指標に関して、同じ図の右側に記載しているような傾向になっているのではないでしょうか。

➲ 閾値に応じた精度評価指標の傾向［図5-4-14］

	モデル精度評価指標			
閾値	Accuracy(正解率)	Precision(適合率)	Recall(再現率)	F1
0	30.6%	30.6%	100.0%	46.9%
0.1	32.2%	31.0%	99.5%	47.3%
0.2	44.5%	34.8%	93.1%	50.6%
0.3	62.8%	43.4%	71.2%	53.9%
0.4	71.3%	53.9%	44.1%	48.5%
0.5	71.7%	60.5%	21.7%	31.9%
0.6	71.0%	69.9%	9.3%	16.4%
0.7	70.0%	72.6%	3.0%	5.8%
0.8	69.6%	100.0%	0.7%	1.3%
0.9	69.4%	0.0%	0.0%	0.0%



Precision と Recall がトレードオフの関係になる

Accuracy（正解率）

閾値を適切に設定することで、上がる

Precision（適合率）

閾値を高くすればするほど、上がる

Recall（再現率）

閾値を低くすればするほど、上がる

F1 スコア

閾値を適切に設定することで、上がる

Accuracy と F1 スコアに関しては閾値を適切に設定することで改善しそうです。ただし、Confusion Matrix 内の値によって適切な閾値の値が変わってくるので、一概にどの閾値とすればよいかはわからず、適用するデータごとに検証する必要があります。

一方で Precision と Recall に関しては明確な傾向があります。結論としては、**Precision と Recall はトレードオフの関係にあり、Precision を上げ（下げ）れば、Recall は下が（上が）ります**。これらの指標の結果を見ながら、どの閾値が適切かを判断することになります。とはいえこのままでは、どの閾値が一番適切そうか、いまいちピンときません。そこでやはり「ビジネス的にどうか？」という観点からも検証してみましょう。

練習用ファイル：chap05_mail_targeting / test_result.xlsx

実践 ビジネス上のKPIをシミュレーションする

回帰問題による需要予測の際は、廃棄数や機会損失数といったビジネス KPI を設定しました。今回も同様に、モデルの精度評価を、ビジネス上の KPI で評価してみましょう。再掲になりますが、あくまでオフライン（机上）での検証であり、いくつかの前提条件をおいて、理論的に KPI を算出および評価するに過ぎません。したがって現実的にどうなるかはもちろんわかりませんし、想定と異なる可能性が十分にあることに留意しておきましょう。それでも、実際にどう成果とする KPI を定義するか？ そして机上とはいえ、ある程度どういったシミュレーション結果になるか？ を事前に見ておくことはとても価値があるので、しっかりと検証しておきましょう。

冒頭の課題設定でも記載していますが、今回は「メール配信による CV とオプトアウトの両方を考慮しながら配信ユーザーを決定し、利益を最大化することを目指す」ことが目標でした。したがって、以下のようなシミュレーションにより、想定される「期待収益」を算出してみましょう。

➲ 期待収益を算出するシミュレーション［図5-4-15］

・仮に、1 回の予約（＝1CV）に対して手数料売上として平均 1,000 円が収益として入るとし、
・また、「実際は予約しないがメール配信を受信したユーザー」の 5% がオプトアウトしてしまい、メール配信による LTV が 10,000 円失われるとしましょう
・その前提において、以下で定義される期待収益を算出する
CV 数 × 手数料 1,000 円 −（配信ユーザー数 − CV 数）× 5% × 10,000 円

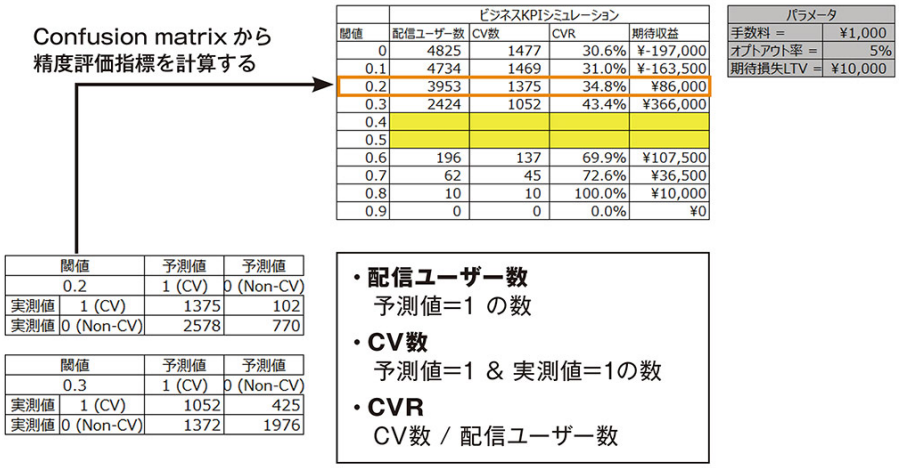
手数料売上 1,000 円、オプトアウト率 5%、期待損失 LTV[※17]10,000 円というのは仮置きですが、実務的にはできるだけ過去のデータから情報を集めて設定しましょう。期待される売上は「CV 数 × 1CV あたり手数料売上」となり、そこから期待損失としての「配信したが CV していないユーザー数 × オプトアウト率 5% × 1 人あたり損失 LTV10,000 円」を差し引くことで、期待収益を算出できそうです[※18]。実際にその計算の結果は、同シートの M 列～ T 列に記載しています。閾値ごとの Confusion Matrix から、配信ユーザー数・CV 数・CVR を［図 5-4-16］のように計算できるでしょう。それらの値から、期待収益を以下のように計算します。

➲ 期待収益の計算式［図5-4-16］

CV 数 * 手数料 −（配信ユーザー数 − CV 数）* オプトアウト率 * 期待損失 LTV

（※ 17）将来見込まれる損失金額と考えればよいです。今回はオプトアウトにより今後メール配信ができなくなるので、ユーザーに有益な情報を与えることによる将来の予約が見込めなくなるので、その金額感を試算した結果だと捉えてください。
（※ 18）ケースによってはこのような計算ではない場合もあるはずです。またメールサービスによってメール配信コストなどがある場合は、そういったコストも鑑みる必要があるでしょう。このあたりの試算は、ビジネスによって異なるので、柔軟に考えるべし、ということを念頭においておきましょう。

➲ Confusion Matrix から各ビジネス KPI を計算する［図5-4-17］



今まで同様に、閾値が 0.4、0.5 の部分を空欄しておいたので、それぞれの KPI を計算してみてください。また全閾値の KPI が計算できたら、モデル精度評価指標と同様に、グラフ化しましょう。「横軸が閾値・縦軸が各ビジネス KPI（左軸が配信ユーザー数と CV 数、右軸が期待収益）」をとったグラフを描画しています（次ページの［図 5-4-18］）。

今回は、基本的には期待収益のみに着目したいです。そして結果を見ると、**閾値が 0.4 ほどのときに、期待収益が最大化されている**ことが見てとれます。したがって、今回は閾値を 0.4 とおき、「予測確率が 0.4 以上のユーザーにはメール配信をする、0.4 未満にはしない」とするのが今回はよさそう、というのがわかりました！[※19]

これにより、実際に配信したいユーザーリストデータがあれば、そのユーザーごとに配信するのかしないのかの予測フラグを付与してあげれば配信実務に落とし込めます。もちろん、予測フラグを用いて自動的に配信するわけではなく、配信管理者が最終確認をするといったケースもあるかもしれません。たとえば何かしらの理由で配信したくないユーザーやユーザー層などがもしあれば、そういったユーザーの配信フラグを変更する、といった対応が考えられるでしょう。

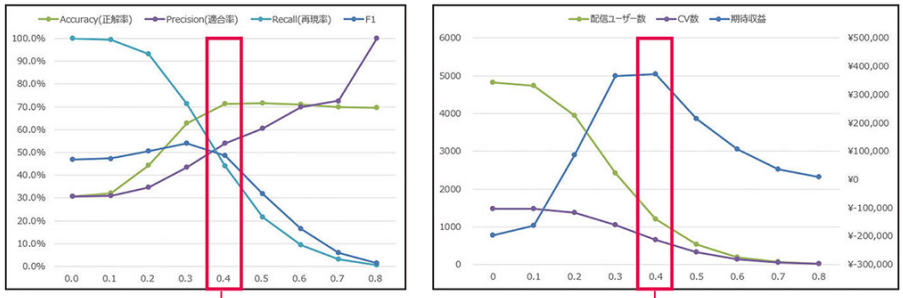
（※ 19）もちろん、そもそものデータ・予測モデル・手数料やオプトアウト率といった数値、などによってさまざまと変わるので注意しましょう。時には閾値が 0.1 や 0.9 がよいときもあるかもしれません。

➲ 閾値ごとのビジネスKPIを計算し、モデルの精度評価指標を照らし合わせる［図5-4-18］

	モデル精度評価指標			
閾値	Accuracy(正解率)	Precision(適合率)	Recall(再現率)	F1
0	30.6%	30.6%	100.0%	46.9%
0.1	32.2%	31.0%	99.5%	47.3%
0.2	44.5%	34.8%	93.1%	50.6%
0.3	62.8%	43.4%	71.2%	53.9%
0.4	71.3%	53.9%	44.1%	48.5%
0.5	71.7%	60.5%	21.7%	31.9%
0.6	71.0%	69.9%	9.3%	16.4%
0.7	70.0%	72.6%	3.0%	5.8%
0.8	69.6%	100.0%	0.7%	1.3%
0.9	69.4%	0.0%	0.0%	0.0%

	ビジネスKPIシミュレーション			
閾値	配信ユーザー数	CV数	CVR	期待収益
0	4825	1477	30.6%	¥-197,000
0.1	4734	1469	31.0%	¥-163,500
0.2	3953	1375	34.8%	¥86,000
0.3	2424	1052	43.4%	¥366,000
0.4	1210	652	53.9%	¥373,000
0.5	529	320	60.5%	¥215,500
0.6	196	137	69.9%	¥107,500
0.7	62	45	72.6%	¥36,500
0.8	10	10	100.0%	¥10,000
0.9	0	0	0.0%	¥0

パラメータ	
手数料 =	¥1,000
オプトアウト率 =	5%
期待損失LTV =	¥10,000



今回は閾値 =0.4 がよさそう

■ ここで学んだ重要トピック

- 教師あり学習、回帰問題
- ロジスティック回帰モデル
- 予測確率、予測フラグ
- Confusion matrix
- Accuracy、Precision、Recall、F1 スコア

■ ステップアップにつながるトピック

- 交差エントロピー誤差関数
- 尤度、対数尤度
- ニュートン法、最急降下法、勾配降下法
- 一般化線形モデル
- 決定木、ランダムフォレストなどによる分類問題
- AUC
- F-beta スコア
- 不均衡データへの対応アプローチ

Chapter 6

ディープラーニングで画像分類を行う

Section

01 画像の商品カテゴリを推測して入力作業を自動化しよう

最近、フリマサイトにはまってるんですけど、品物の細かいデータを入力するのが面倒なんですよね。

ちょっと見せてみて……。あー、商品画像をアップしてからカテゴリを選んで、さらにそこから値段などを設定していく必要があるのね。たしかにこれでは出品の途中で離脱してしまう人も多そうね。

そうなんです。最近は AI で画像認識もできるんだから、アップした写真くらい自動認識してカテゴリ分けしてほしいですよね。

いいわね！ ではそのフリマサイトに提案書を持っていこう！

え？ 何の話ですか？

決まってるでしょ？ そのフリマサイトの運営会社に、画像認識による出品手続き簡素化ソリューションを提案しにいくのよ。出品も楽になって小遣い稼ぎもしやすくなるし、あなたのお給料アップにもつながるし、一石二鳥ね！

ここで学ぶこと

- ☑ 画像データの取り扱い
- ☑ 画像分類を実務で活用するための考え方
- ☑ ニューラルネットワーク、DNN、CNN の仕組み

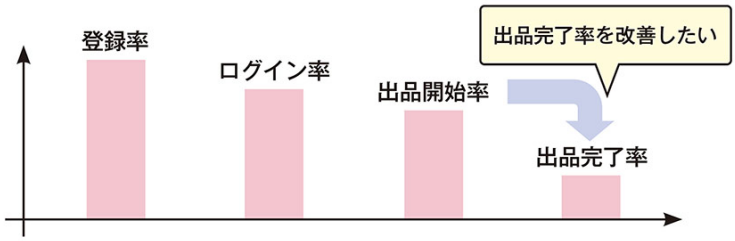
とあるフリマサイトの課題を考えてみよう

とあるフリマサイトの運営会社Ｄ社のケースを考えてみましょう※1。「フリマサイト」とはメルカリ、ヤフオク、ラクマといったネット上のフリーマーケットサービスのことを指します。Ｄ社のサービスサイトに登録したユーザーには主に「出品」と「購入」の２つの行動パターンが存在し、商品の出品者と購入者がフリマサイト上でマッチングすることで商品の売買が成立します。Ｄ社はその売買手数料が主な収益源となっています。

➲ 出品者と購入者をマッチングさせるプラットフォーム［図6-1-1］

したがって、より多くのユーザーが商品を出品したり、商品を購入したりしてくれることが、Ｄ社のビジネスにとって重要となります。そこで現状を分析したところ、特に出品数が少ないことが浮き彫りになりました。さらにユーザーがサイトに登録してから実際に出品が完了するまでの行動を分解し、KPI を観察しました。すると出品を試みてから完了するまでに比較的多くのユーザーが離脱してしまい、結果的に出品完了率が低くなっているという課題が見つかりました。そこでＤ社は、できるだけユーザーが簡単に出品できるように、さまざまな施策を講じることにしました。

➲ 出品者の出品完了率を改善したい※2［図6-1-2］

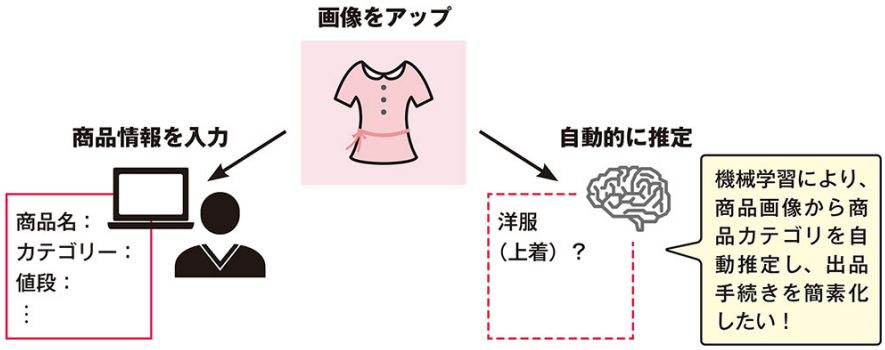


（※1）本章もわかりやすい例として当該サービスをもとに考えていきますが、EC サイトや製造業など、画像を取り扱うような事業やサービスでも、もちろん同じような活用が考えられます。
（※2）このような消費者の購買行動プロセスを分解して、それぞれのプロセスにおける KPI を測定・分析する方法を「ファネル分析」といいます。

そのうちの１つの施策として、出品者ができるだけ手間暇かけずに出品を完了できるように、**出品したい商品画像をサイトにアップロードしたら、自動で商品カテゴリを推定することで、手入力の工数を減らし、出品手続きを簡素化しよう**と考えました（商品画像から商品名を推測したり最適出品価格や配送料を提示したり、といった施策も考えられます。技術的にも実現可能性はありますが、今回は商品カテゴリの推測にスコープを絞ります）。

そこで、出品者からアップロードされた過去の商品画像と、各画像に紐づく商品カテゴリのデータを利用し、画像を用いた機械学習モデルを構築することで、本施策の実現を目指します。

➲ 商品カテゴリを画像から推測し、入力手続きを簡素化したい［図6-1-3］



データサイエンスで解くための問題設定

問題設定をもう少しだけ詳細化します。画像解析といっても、どう画像を解析するかという解き方は非常に多岐に渡ります。今回の問題は、

「その画像は、どの画像カテゴリ※3 に分類されるか？」

という問題が解ければよく、画像解析のうちの「画像分類」に相当します。画像解析において画像分類は一番基本的かつ重要な問題設定なので、本書でしっかりイメージをつかみましょう。

大量の画像データ（インプット）と画像カテゴリデータ（アウトプット）

（※3）カテゴリと同義語として「ラベル」や「クラス」という用語もあります。ここでは、一律に「カテゴリ」として進めます。

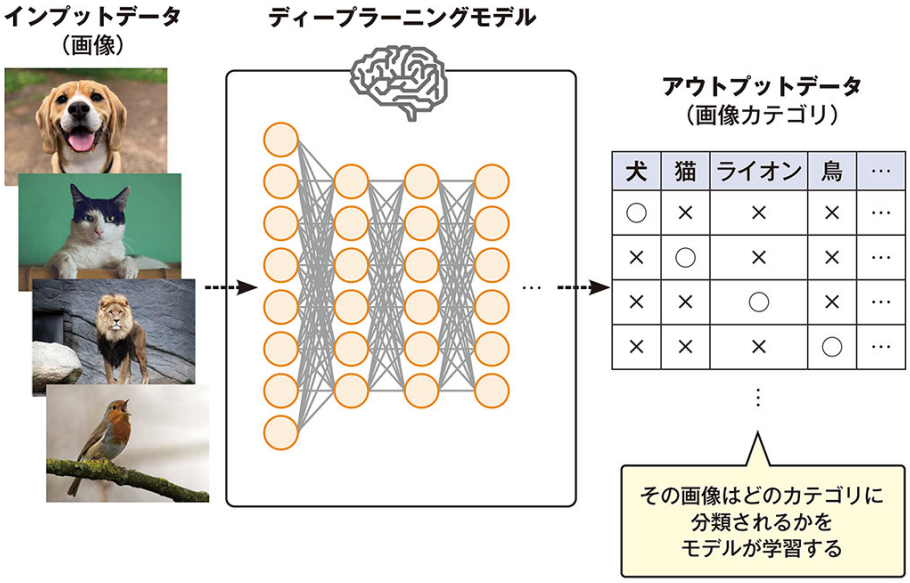
をモデルが学習し、新たな未知の画像がきた際に、その画像はどのカテゴリの画像なのかを、高い精度で予測させることがゴールとなります。

対象とする画像データを決め、それらをどう分類するかという「カテゴリ」を定義し、画像ごとにラベリングします。たとえばこの画像は「犬」、この画像は「猫」……といった具合です。今回は、過去の登録商品の画像データおよび画像カテゴリのデータを用いることとなります。

仮に画像のカテゴリが「犬・猫・ライオン・鳥……」で100種類であれば、それは100カテゴリの分類問題となります。つまり、第4章でやった分類問題は、「するかしないか」（Yes/No）の二値分類でしたが、今回は複数カテゴリがアウトプットとなる**多値分類**問題となります。

そして、ある画像がどの画像カテゴリかを、ディープラーニングモデルに学習させます。その結果、ある新たな画像に関して、その画像は何のカテゴリなのかを予測（推論※4）できるようになります。

➲ 画像と画像カテゴリのデータを用いて画像分類問題を解く［図6-1-4］



（※4）画像解析の場合はしばしば「推論」（Inference）というワードが使用されることがあります。基本的な使い所は「予測」とほぼ同義と考えて差し支えありません。違いとしては、予測というと「将来の情報」を推測するといった意味合いが強い一方で、画像解析の場合は将来情報という意味合いが弱く、単に未知の画像からカテゴリを推測するといった意味合いが強いため、推論（既知の事柄をもとにして未知の事柄について予想する）という言い方をすることが多いと考えられます。

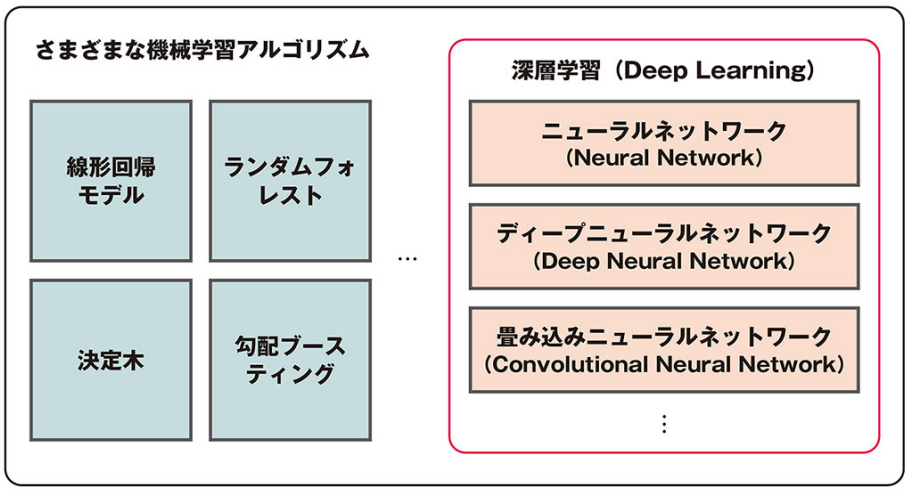
Section

02 ディープラーニングの基本「ニューラルネットワーク」

画像解析に適用する機械学習アルゴリズム

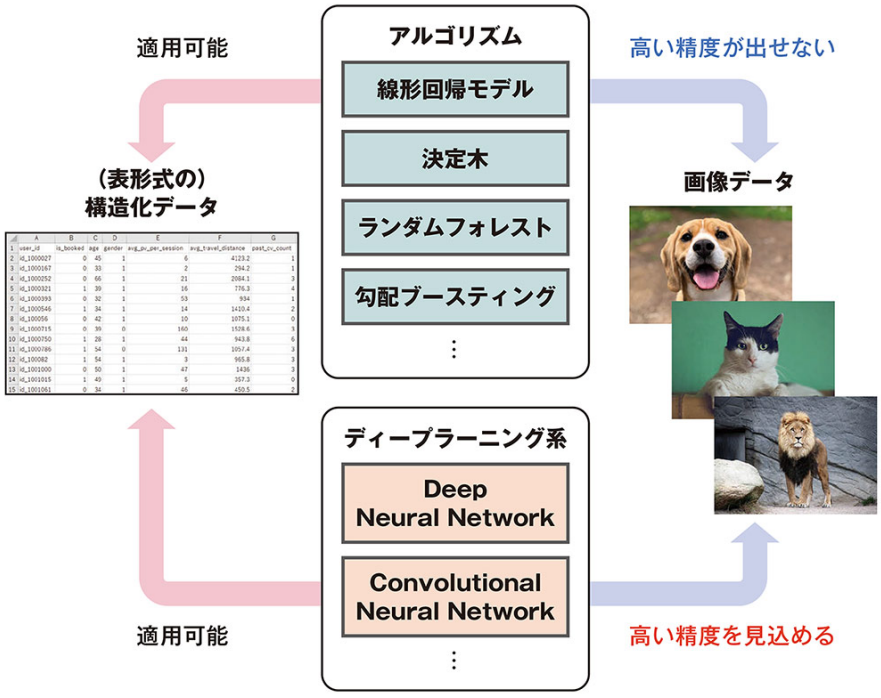
それでは、具体的に画像分類を解くための技術概要を押さえていきましょう。昨今、画像解析の技術は日進月歩の勢いで進化していますが、その要因は**「深層学習」（Deep Learning、ディープラーニング）**の発展にあります。そもそもディープラーニングというのは、これまで紹介したような線形回帰モデルやロジスティック回帰モデルといったものと同様、さまざまな機械学習アルゴリズムの一部分を指します。ディープラーニングにも、この後紹介するニューラルネットワークや畳み込みニューラルネットワークといったさまざまなアルゴリズムが存在します。近年では非常に多くのアルゴリズムが開発されているため、本書ではほんの一部しか紹介できませんが、冒頭に紹介した物体検出・姿勢推定・画風変換……といったさまざまな問題設定に応じて、適切なアルゴリズムの選択が必要になります。

➲ ディープラーニングは機械学習アルゴリズムの一部である［図6-2-1］



ではなぜ画像解析の文脈でディープラーニングを取り上げるのでしょうか。実はディープラーニングのさまざまなアルゴリズムは、これまでのような表形式のデータ(「構造化データ」といわれます)にも適用できます。そして、線形回帰モデルやロジスティック回帰モデルといったアルゴリズムは、画像解析に対しては高い精度が出せないことが知られています。一方で、ディープラーニングの各種アルゴリズムは画像データに対して高い精度が見込めるようになってきています。そして拍車をかけるように、画像解析に特化したディープラーニング系アルゴリズムがどんどん開発されているため、**画像解析をするのであればディープラーニングのアルゴリズムを利用することがほぼデファクトスタンダード**となっています※5。

➲ 画像解析においてディープラーニングは高い精度が見込める［図6-2-2］



そこで、画像解析の中でも基本中の基本アルゴリズムであるニューラルネットワーク、ディープニューラルネットワーク、畳み込みニューラルネットワークをどう画像分類問題に適用させるかの概要を学んでいきましょう。

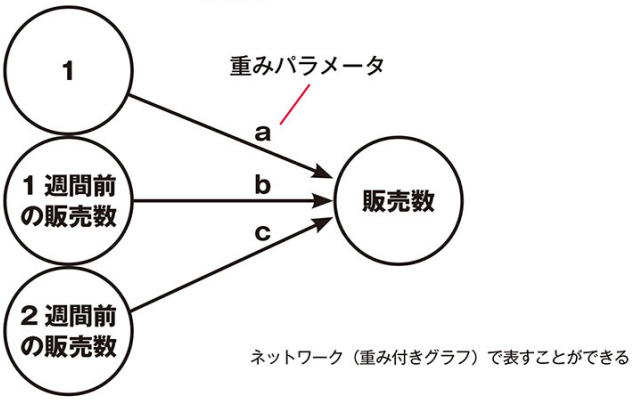
(※5) 補足ですが、ディープラーニング系のアルゴリズムは非常に発展が進んでおり、テキストデータに関連する複雑な問題（自動翻訳や文章生成・要約など）でもよく利用されるようになってきています。

線形回帰モデルをネットワーク構造で表す

まずは先ほど学んだ線形回帰モデルを別の視点で捉えてみましょう。線形回帰モデルは特徴量（インプット）と目的変数（アウトプット）を直線（線形）の関係式で定義したモデルでした。そのモデル式は、見方を変えると［図6-2-3］のような**ネットワーク構造**で表すことができます。ネットワーク構造というのは、点と点が辺でつながったようなグラフ構造のことを指しますが、特徴量ごとに回帰係数（傾き）をかけ合わせることで目的変数（図における販売数）を計算する構造を示しています。回帰係数は点（特徴量）と点（目的変数）をつなぐような構造になっており、このようなネットワークは**「重み付きグラフ」**と呼ばれ、ネットワーク構造において回帰係数は**「重みパラメータ」**と呼ばれます（本質的な意味は回帰係数と変わりません）。

➲ 線形回帰モデルをネットワーク構造で表す※6 ［図6-2-3］

線形回帰モデル：
販売数 ＝ a ＋ b × 1 週間前の販売数 ＋ c × 2 週間前の販売数 ……



ネットワーク（重み付きグラフ）で表すことができる

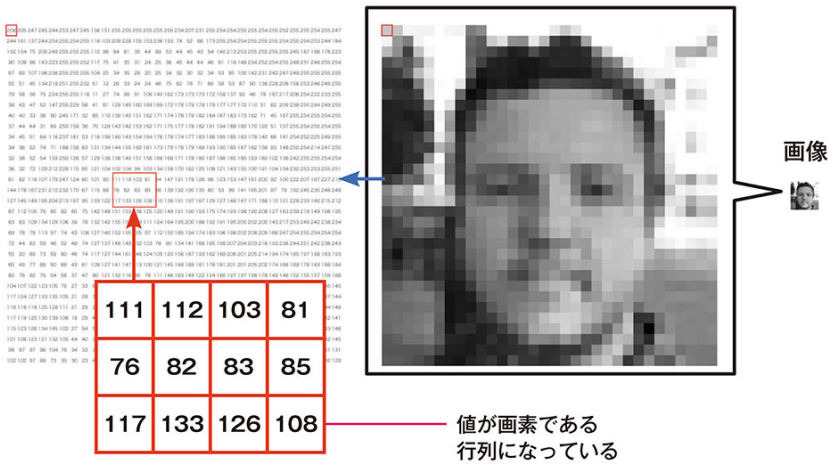
画像データは行列データである

線形回帰モデルがネットワーク構造で表せることがわかったことで、ニューラル「ネットワーク」の概念に少し近づきました。もう１つ重要なポイントを押さえておきましょう。

（※6）細かいですが、切片 a もネットワークでうまく表すために、1 という点を作ることで、販売数 ＝ 1 × a ＋ 1 週間前の販売数 × b ＋ … と表現しています。

それは、**画像データは実は行列データとして表せる**ということです。画像は、**画素が１つ１つ並ぶことで構成されています**。よく画像が「横 700px × 縦 500px」などと表されていますが、あれは 700 × 500 の画素値で構成されている画像ということを示しています。またその画素値は 0 から 255（0 に近いほど黒、255 に近いほど白）の範囲で定義されます。つまり横 700px × 縦 500px の画像というのは、単に各要素に 0 ～ 255 の値が格納されている 700 × 500 の行列データである、と捉えることができます※7。

➲ 画像データは行列データである［図6-2-4］



出典：https://setosa.io/ev/image-kernels/

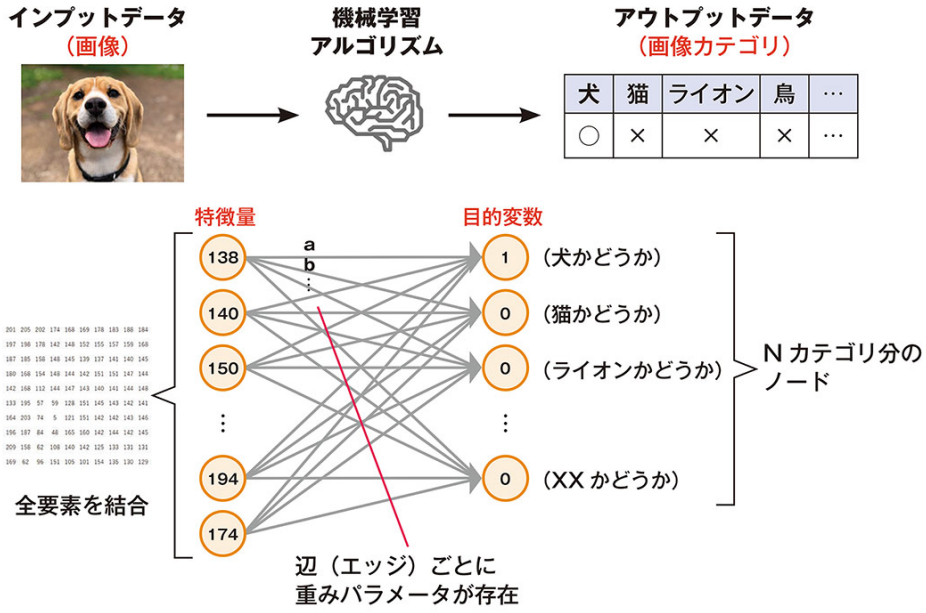
画像の分類問題をネットワーク構造で表す

以上を踏まえ、画像データから画像カテゴリを学習するモデルを考えます。結論として画像の分類問題は［図 6-2-5］のようなネットワーク構造で捉えられます。インプットとなる画像データは行列データであり、**行列内の全要素を１列に結合する**ことで、線形回帰モデルと同様のインプット構造にできます。つまり、**行列内の画素値を特徴量にする**ということです。さらにアウトプットとなる画像カテゴリに関しては、仮にカテゴリ数が N 種類なら N 要素の点（ノード）をアウトプットとして作成し、その**画像が該当するカテゴリに 1、それ以外に 0** を付与します。そしてインプットの特徴量として

(※7) 私たちに馴染みの深いカラー画像はRGBカラーモデルの場合が多く、このときはRed、Green、Blueそれぞれ1つずつ、合計3つの行列が存在していると考えることができます。

の画素値とアウトプットの目的変数としてのカテゴリに関して、全組み合わせの辺（エッジ）をつなぐことでネットワークが完成します。またこのエッジには、線形回帰モデルの回帰係数と同様に重みパラメータが存在します。

➲ 画像の分類問題をネットワークとして捉える [図6-2-5]

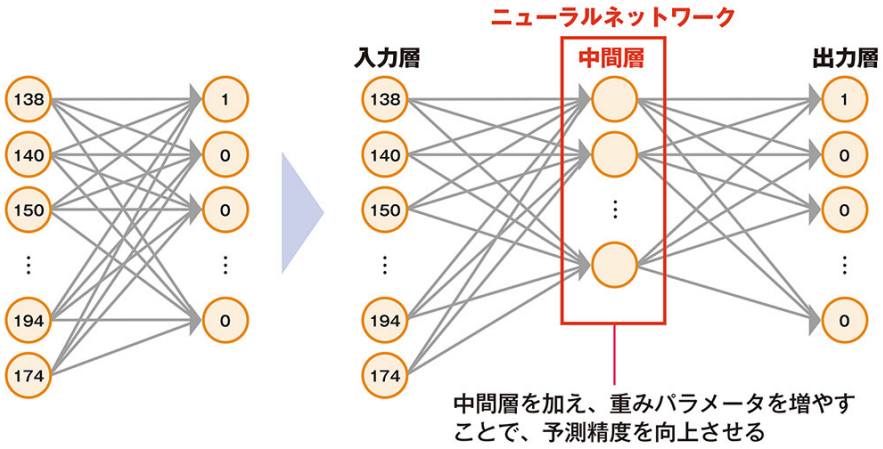


中間層を加えたニューラルネットワーク

ここまでで画像の分類問題をネットワーク構造で捉えることができました。ただしこのままだと、画像の特徴量から目的変数までエッジが１回しか存在せず、モデルとしては線形回帰モデルやロジスティック回帰モデルと同程度で精度が上がりません。そこで、「入力層」としての特徴量と「出力層」としての目的変数の間に**「中間層」**を加えます（図 6-2-6）。中間層には任意の数の点（ノード）を加えることができます。そして入力層から中間層、中間層から出力層にかけて、それぞれの辺（エッジ）に重みパラメータを加えたものが**「ニューラルネットワーク」（Neural Network）**と呼ばれます。

入力層の特徴量が、中間層のノードと付随する重みパラメータを介することでより複雑な計算ができるため、予測精度を向上させられます。

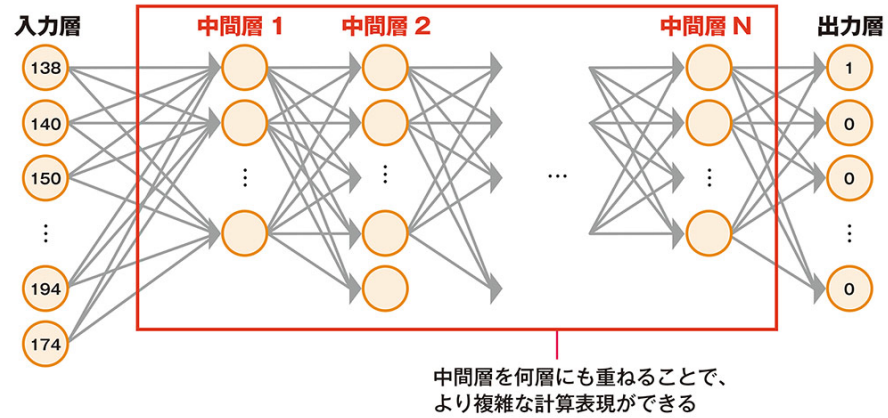
➲ 中間層を加えることでニューラルネットワークが構築できる [図6-2-6]



ニューラルネットワークを多層化した「DNN」

先ほどは中間層を 1 層加えましたが、実はこの中間層は何層でも増やすことができます。［図 6-2-7］のように、中間層をたくさん重ねたネットワークは**「ディープニューラルネットワーク」（Deep Neural Network、DNN）**と呼ばれます※8。中間層を何層にも重ねることにより計算表現を複雑にでき、線形回帰モデルなどでは精度高く予測するのが難しい場合でも、より高い精度で予測するモデルが構築できる可能性が高まります※9。

➲ 中間層を何層も加えたディープニューラルネットワーク [図6-2-7]



（※ 8）中間層を何層加えるかというのは任意の数で指定できます。一般的には、層を加えるほど、その分学習データが必要となりますが、適切に学習ができれば精度は高くなる傾向にあるといわれています（もちろん必ずではないですが）。
（※ 9）もちろんケースバイケースで、線形回帰モデルでも DNN と同程度の精度が出るようなこともあるので、あくまで傾向として、DNN は線形回帰モデル等よりも精度が高く出やすいということに留意してください。

なお、このような中間層のいくつものノードを介して入力層から出力層まで計算されている様子が、シグナル伝達などの神経活動により情報伝達がされる脳機能の特性に類似していることから「ニューラル」と呼ばれています。

「学習」により重みパラメータを最適化

これで、DNN のネットワーク構造を把握できました。実務的には、ネットワークを定義したら、あとは大量の学習データを用意し、ネットワークに学習させます。ここでいう学習とは、**たくさんある辺（エッジ）に紐づくすべての重みパラメータの値を最適化する**ことを指します。重みパラメータは最初はテキトウな初期値になっていますが、たとえばある画像の正解が犬カテゴリであるということを適切に出力できるように、重みパラメータの値をより適切な値に修正していくというプロセスとなります※10。

［図 6-2-8］を見て何となくイメージできると思いますが、中間層を増やしたり、各中間層のノード数（頂点の数）を増やしていくと、指数関数的にエッジの数が増え、重みパラメータの数が増えます。したがって、DNN はより学習に時間がかかる、正確にいえば**ネットワークが複雑になればなるほど学習に時間がかかる**、という部分だけ押さえておきましょう。

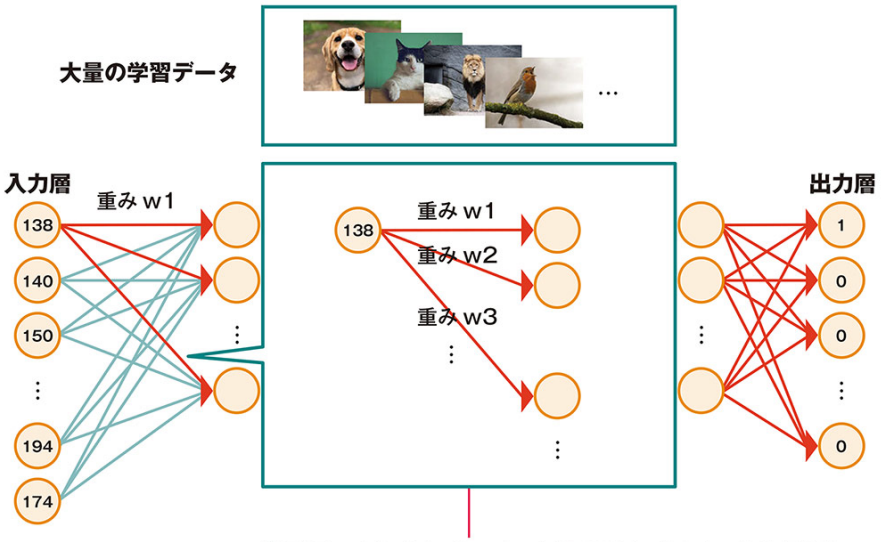
学習したネットワークを利用した「予測」

重みパラメータを学習したネットワークができれば、予測（推論）可能です。新しい未知の画像があった場合、その画像の行列データ（画素値）を抽出して入力層として、ネットワークに沿う形で出力層まで値を計算し続けます。その際にエッジの重みパラメータは学習された値を利用します。

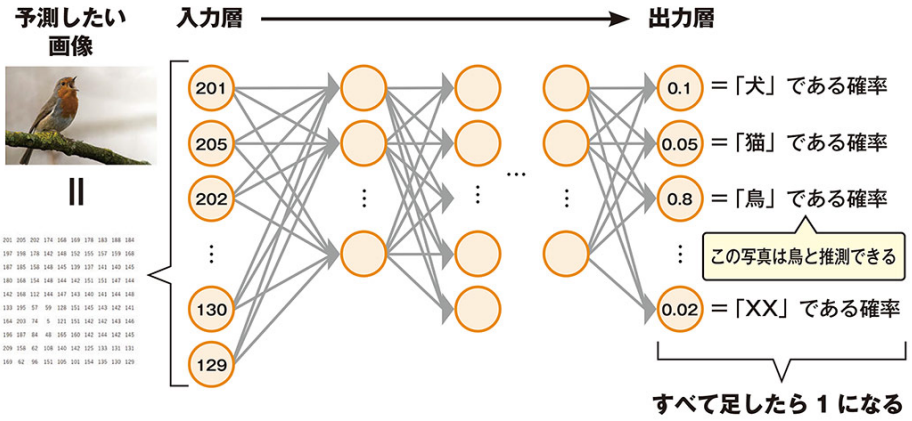
最後の出力層ですが、ここでは予測確率を出力します。今回はカテゴリ数が N 種類ある「多値分類」問題なので、「犬である確率」「猫である確率」……と N ノードの出力層それぞれに予測確率が出力され、**すべてのカテゴリの予測確率を足し合わせると 1 になる**ようになります※11。そして、**予測確率が最も高いノードに相当するカテゴリが、その画像の分類されるべきカテゴリである**と判断できます。

（※ 10）具体的にどのように重みパラメータを学習するか？という部分は、数学的に少し難解な説明となってしまいますので、本書では説明を省きます。技術的には、「勾配降下法」や「誤差逆伝播法」という方法論を利用して学習します。

➲ ネットワークの学習＝重みパラメータを最適化すること [図6-2-8]



➲ 未知の画像から画像カテゴリの予測確率を出力できる [図6-2-9]



学習したネットワークを利用して、未知の画像からカテゴリの予測確率を出力する

(※11) なお、最後に足し合わせると1になるようにするために、「ソフトマックス関数」と呼ばれる関数が入れ込まれています。これはロジスティック回帰モデルにおいて同じく予測確率を出すためのシグモイド関数の多カテゴリ版に相当します。

Section

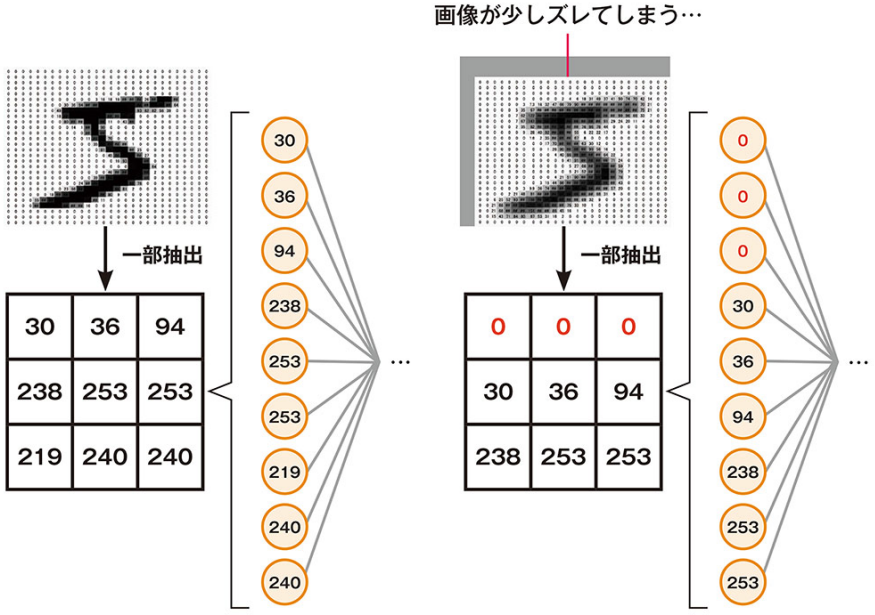
03 画像認識のための「CNN」

画像解析におけるDNNの問題点

これでDNNにより画像の分類問題を解ける状態になりました。しかし、実は**DNNは画像解析には向いていない**ことが知られています。それはなぜかというと、画像データは、被写体や背景といった情報が統一的に同じような箇所に固定されているわけではなく、「ズレ」が生じてしまうためです。**画像に少しでもズレが生じてしまうと、インプットとしての入力層が大きく異なってしまうために、高い精度で学習できない**という問題があります。

したがって、今まで紹介してきたDNNを、画像のズレなどの影響を受けにくいような形に改善する必要があります※12。

➲ 画像にズレが生じると、入力層の値がまったく異なってしまう［図6-3-1］



出典：http://yann.lecun.com/exdb/mnist/

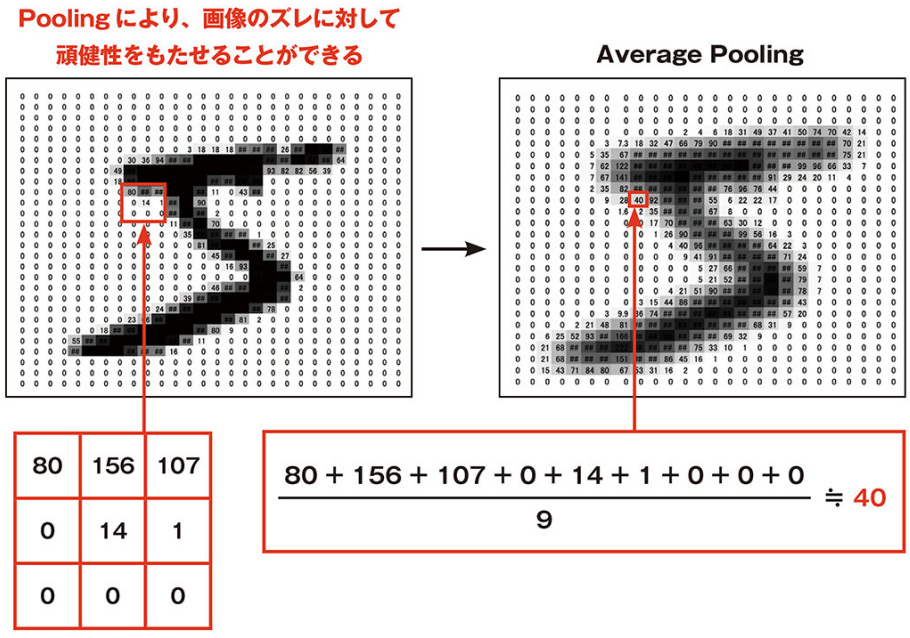
（※12）補足すると、DNNの構造だと、画像のズレ以外にも、隣り合う要素同士の影響を加味することができない（独立して扱ってしまう）という問題点もあります。当たり前ですが、画像では隣り合う画素同士が“いい感じに”描画されているために画像として認識できる構造になっているため、その隣合う要素をうまく情報として抽出したい、という点もDNNを改善すべきポイントとなります。

画像のズレを吸収する「プーリング（Pooling）」

画像のズレを吸収する１つの方法として、**画像をぼかすことでズレを吸収する**というやり方が考えられます。その具体的なやり方として、［図 6-3-2］にあるプーリング（Pooling）という方法、特に今回は **Average Pooling** を紹介します。考え方は簡単で、画像におけるある範囲（2 × 2 や 3 × 3 など）の画素値の「平均値」を計算します※13。そして画像内のすべての範囲で同様の計算をすることで、画像を全体的に少しぼかします。少しモザイクがかかっているようなイメージですね。範囲内の平均を取ることで、画像にズレがあっても吸収できるということです。

実務的には、Average Pooling のほかにも、範囲内の「最大値」を計算する Max Pooling といったやり方も存在します。

➲ Pooling により、画像をぼかしてズレを吸収する［図6-3-2］



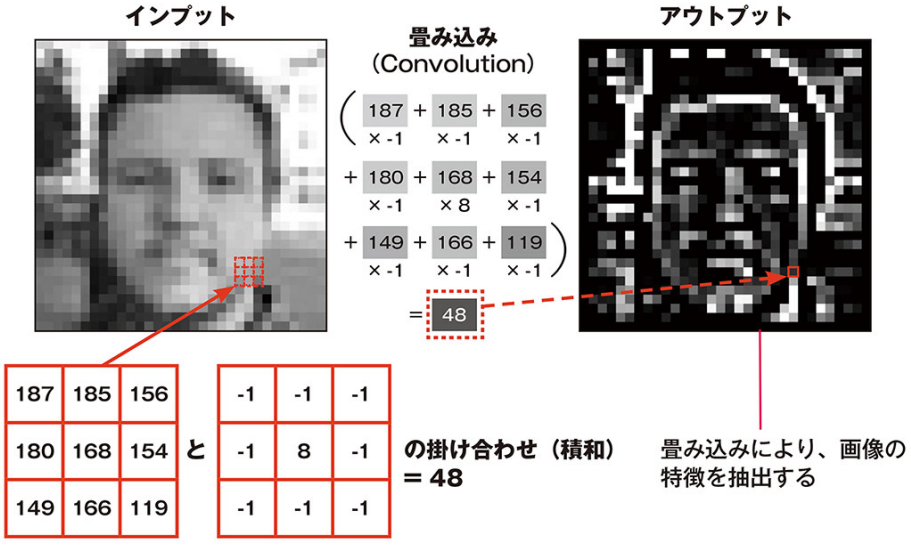
出典：http://yann.lecun.com/exdb/mnist/

(※13)範囲を 2 × 2 にするのか 3 × 3 にするのか…といった指定は任意の範囲とすることができます。さまざまなネットワークにより異なりますが、2 × 2 の Pooling をすることが多い印象です。

画像の特徴を抽出する「畳み込み（Convolution）」

Pooling のほかに、もう 1 つ面白い技術があります。先ほどの Pooling は画像のズレを吸収する役割を果たしていますが、ここで紹介する「**畳み込み（Convolution）**」という概念は、ズレを吸収するほかに、**画像の特徴をうまく抽出する**という役割を持っています。単にある範囲内で平均値や最大値を取るだけではなく、［図 6-3-3］のように、ある行列を別途用意して、画像の範囲内の行列と掛け合わせる（積和を計算する）ことで、画像から、特徴的な部分が抽出された画像（行列）に変換できます。図では、まさに顔の輪郭部分がうまく取り出されていそうですね。このように、いろいろな画像があったとしても、それらの画像から、たとえば輪郭部分だけをうまく抽出することで、人間らしい輪郭、犬らしい輪郭……といった特徴をネットワークに適切に学習させられるようになります。なお、図にある（-1, -1, -1, -1, 8, -1 -1 -1 -1）の行列をカーネルあるいはフィルタと呼びます。

➲ Convolution により、画像の特徴を抽出する［図6-3-3］



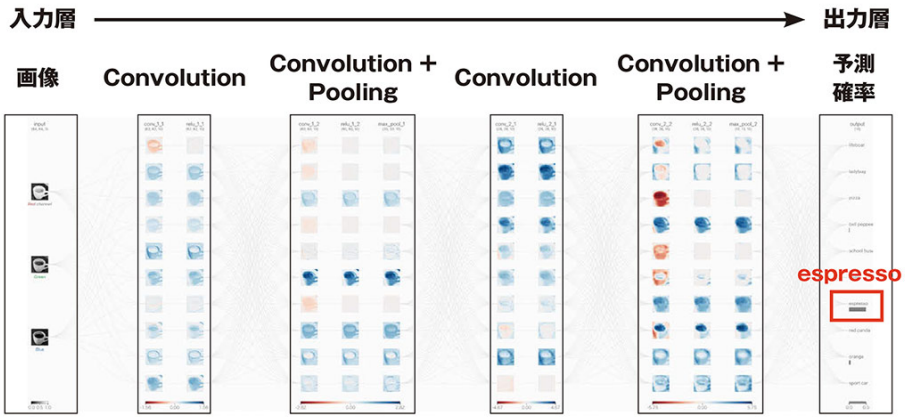
出典：https://setosa.io/ev/image-kernels/

画像認識の精度が高い「CNN」

Pooling や Convolution を紹介しましたが、これらのパーツをたくさん組み合わせたネットワークを**「CNN」（Convolutional Neural Network）**と呼びます。先ほど紹介した DNN と大きな構造は似ていて、入力層から出力層にかけてたくさんの中間層を組み込んだ複雑なネットワークですが、CNN では、中間層の部分に Convolution 層や Pooling 層も加えます。そして［図6-3-4］のように Convolution 層や Pooling 層をたくさん加えてネットワークを構築し、**画像のズレを吸収したり、画像の特徴を適切に抽出したりするというプロセスを経ることで、より高い精度で画像のカテゴリを分類できる**ようになります。

大量の画像データをもとに、先ほど紹介したカーネルと呼ばれる行列の値や、DNN で紹介した重みパラメータを学習し、ネットワーク全体を学習させます。

➲ CNN（Convolutional Neural Network）のイメージ［図6-3-4］



Convolution や Pooling をたくさん加えてネットワークを構築することで、精度よく画像分類が可能

出典：https://poloclub.github.io/cnn-explainer/

Tips　学習済みモデルを利用する

近年では、ディープラーニングやコンピュータ性能などの技術発展もあいまって、さまざまなネットワークが構築されています。ここ数年では、非常に複雑なネットワークが開発されており、先ほど学んだ重みパラメータが数千〜数億個も存在するようなネットワークもざらにあります。そのような大規模なネットワークは学習するためにとても多くの画像データが必要になり、学習自体にとても多くの時間がかかります。したがって、ある程度の規模になると、相当量のコンピュータリソースが必要になり、自前ですべて学習させるのが難しくなってきています。

一方で、GAFA のような企業が開発・学習させた大規模なネットワークは、しばしばネット上に公開されているケースが多く、そのモデルをダウンロードすれば私たちでも使えます。つまり、大量の重みパラメータがすでに学習された状態のモデルを使うことができるのです！

また、そのような学習済みモデル（Pre-trained model）は、ある程度一般的な画像データを中心に学習させていることが多いため、自分たちで適用したい画像がそれらとは異なる種類の画像である場合は、少しだけ学習済みモデルをチューニングさせる必要があります。このように、学習済みモデルを少しだけ自分たちのデータにチューニングする方法を「**転移学習**」や「**ファインチューニング**」と呼んだりします。チューニングはするものの、大規模なネットワークすべてをゼロから学習し直すわけではなく、すでに学習済みの重みパラメータを利用しつつ、一部分だけ再学習することにより、少ない計算コストで学習できます。それにより、大規模なネットワークの恩恵を受けつつも、自分たちのデータに最適なディープラーニングモデルを手にすることができるようになります。このような方法を利用するケースが近年は多くなってきているので、頭の片隅に入れておくとよいかもしれません。

Section

04 実践：洋服の画像データを活用しよう

練習用ファイル：chap06_image_classification / train_sample/、test_sample/

実践 データの確認

最後に、実践演習を通して、画像解析、特に画像の分類問題のビジネス適用のイメージを深めていきましょう。冒頭で述べたように、今回の検討施策は**出品したい商品画像をサイトにアップロードした際に、自動で商品カテゴリを推定することで、手入力の工数を減らし、出品手続きを簡素化する**ことです。そこで画像分類問題を解くためのデータが必要となります。

本来であれば過去蓄積してきた全画像データを用いるのがベストですが、画像数が多く、また画像の種類も多いため、いきなり全数は取り扱いが難しいと判断しました。そこでまずは**スモールスタートで始めることとし、洋服の画像の一部のみを対象**とし、モデリングを試みることとする、と仮定しましょう。今回使用したデータは「chap06_image_classification」フォルダ内に格納してあります※14。今回扱うデータは画像（JPEG ファイル）となりますので、少し扱い方が特殊です。今回は以下の 2 つのデータセットを使用します。

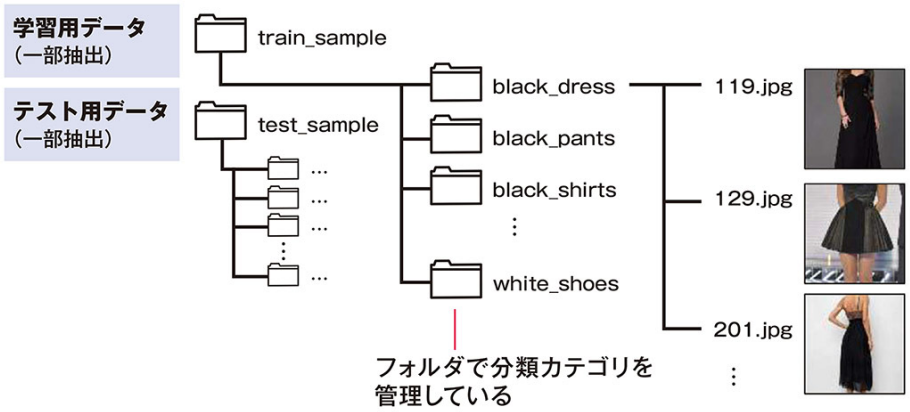
1. モデルを学習するための「学習データ」
2. 学習したモデルの精度を評価するための「テストデータ」

それぞれ、train_sample と test_sample の 2 つのフォルダに画像データを格納しています。また画像データということもあり、元々の枚数だと容量が多くなってしまうので、今回は一部だけ抽出する形としています。その一部抽出した学習用データと、同じく一部抽出した精度評価のためのテスト用データ（train_sample フォルダと test_sample フォルダ）には、今回対象とする 20 カテゴリのフォルダが存在しており、**各フォルダ内に、そのフォル**

（※ 14）今回使用したデータは以下を参考にしています。
https://www.kaggle.com/trolukovich/apparel-images-dataset

ダ名が正解の分類カテゴリとなっている画像を管理・格納しています。今回使用する画像のカテゴリは、洋服の種類や色の種類をもとに判別したカテゴリとなっています。

➲ **使用するデータセットの一覧とフォルダ構成**※15 [図6-4-1]

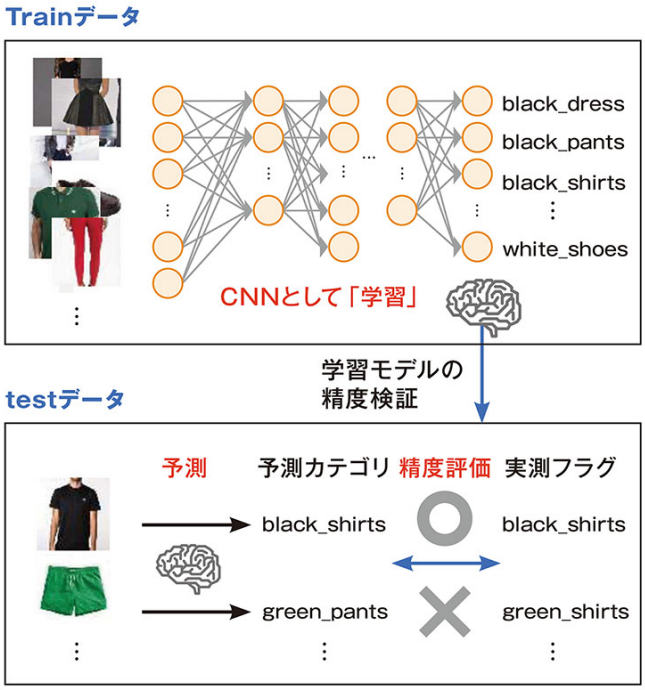


全体の処理の流れを理解する

これらの画像データを用いた全体の処理としては、データセットと同じく3つの処理に分けられます。まずは**正解の分類カテゴリがある学習用データで、ディープラーニングネットワークを学習**させます。今回は上述したCNN（Convolutional Neural Network）として学習してみます。細かいネットワーク構造は省略しますが、Convolution層とPooling層をいくつか重ね合わせ、最後にDNN層を組み合わせた、比較的シンプルかつ一般的なCNNを構築しています。したがって、CNNのネットワーク構造をより複雑にしたり、172ページのTipsで紹介した大規模モデルを転移学習させたり、といったことをすれば、より高精度なモデルを構築することはできるでしょう。次いで、**学習したCNNのモデルと検証データの画像を用いて画像カテゴリを予測し、実測のカテゴリと比較して精度評価**をします。

（※15）画像ファイル名の数字には特に意味はないので気にせずとも大丈夫です。

➲ 全体の処理の流れ [図6-4-2]



実務的には、ユーザーがアップロードした画像データに対して、予測モデルを適用し、その画像のカテゴリを予測します。また多くの場合はシステム連携をし、**得られた予測カテゴリをフリマサイト上に出力し、ユーザーに確認させる**（必要に応じて編集ができるなどの機能も加える）といった流れとなります。今回はシステム機能の実装まではしないため割愛しますが、実務的にそういった後続の処理があることを意識しておきましょう。

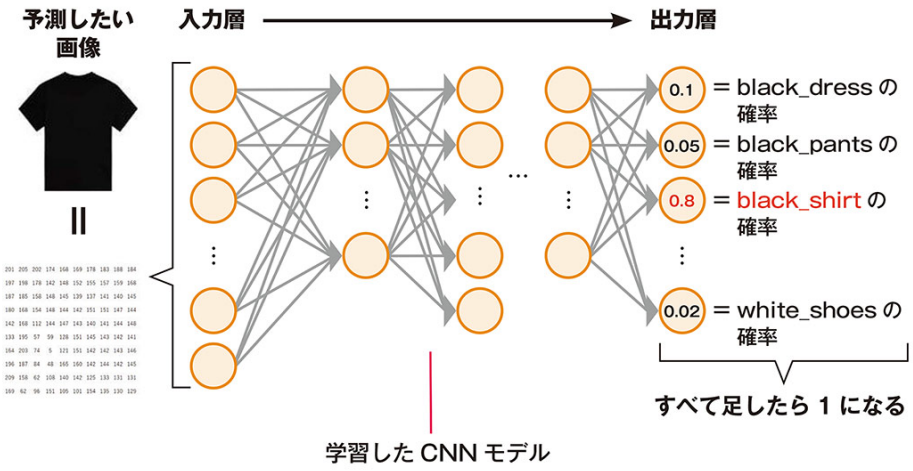
練習用ファイル：chap06_image_classification / test_result.xlsx

実践 学習したCNNによる予測結果を確認する

今回は、約1万枚の画像データ（train_sample フォルダにはその一部の画像が存在しています）を利用して CNN を学習させました。今回は全部で 20 カテゴリの画像データとなっているので、**20 カテゴリの多値分類**となります。その学習させたモデルを、テスト用の画像データ 1,489 枚（test_sample

フォルダにはその一部の画像データが存在しています）に対して適用させました。各画像を学習したモデルに入力させると、20 カテゴリそれぞれの予測確率が出力されます。予測確率が高いカテゴリほど、その画像カテゴリである可能性が高いので、**最も予測確率が高いカテゴリを予測結果（予測カテゴリ）**とします。

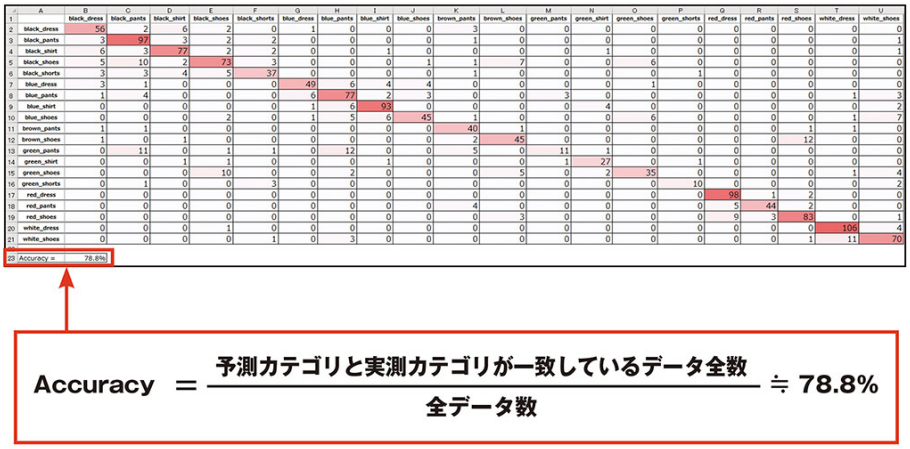
➲ 最大の予測確率をとる画像カテゴリを予測結果とする [図6-4-3]



そのようにして出力されたすべての検証用データの予測カテゴリと、正解となる実測カテゴリを比較した精度評価結果を「test_result.xlsx」に記載しています。1 つ目の「Confusion Matrix」シートには、分類問題を取り上げた際にも登場した Confusion Matrix（混同行列）の計算結果を図示しています。今回は 20 カテゴリあるため、20 × 20 の行列となっていますが、本質的な見方は最初に紹介した 2 × 2 の Confusion Matrix と同様です。**行に実測カテゴリ、列に予測カテゴリ、各要素に該当するデータ数**となっているため、**斜め 45 度線上に存在するデータは予実カテゴリが一致している**と定義することができ、その数が多いほど精度がよいと考えられます。

つまり、**全データ数に対して、斜め 45 度線上に存在するデータ数の割合は、Accuracy（正解率）**となります。その計算を Excel でしてみると約 78.8% であることがわかります。

➲ 画像分類問題の Confusion Matrix（実測 20 カテゴリ×予測 20 カテゴリ）[図6-4-4]



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1		black_dress	black_pants	black_shirt	black_shoes	black_shorts	blue_dress	blue_pants	blue_shirt	blue_shoes	brown_pants	brown_shoes	green_pants	green_shirt	green_shoes	green_shorts	red_dress	red_pants	red_shoes	white_dress	white_shoes
2	black_dress	56	2	6	2	0	1	0	0	0	3	0	0	0	0	0	0	0	0	0	0
3	black_pants	3	97	3	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
4	black_shirt	6	3	77	2	2	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1
5	black_shoes	5	10	2	73	3	0	0	0	1	1	7	0	0	6	0	0	0	0	0	0
6	black_shorts	3	3	4	5	37	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
7	blue_dress	3	1	0	0	0	49	6	4	4	0	0	0	0	1	0	0	0	0	0	0
8	blue_pants	1	4	0	0	0	6	77	2	3	0	0	3	0	0	0	0	0	0	1	3
9	blue_shirt	0	0	0	0	0	1	6	93	0	0	0	0	4	0	0	0	0	0	0	2
10	blue_shoes	0	0	0	2	0	1	5	6	45	1	0	0	0	6	0	0	0	0	1	7
11	brown_pants	1	1	0	0	0	0	0	0	0	40	1	0	0	0	0	0	0	1	1	0
12	brown_shoes	1	0	1	0	0	0	0	0	0	2	45	0	0	0	0	0	0	12	0	0
13	green_pants	0	11	0	1	1	0	12	0	0	5	0	11	1	0	0	0	0	0	0	0
14	green_shirt	0	0	1	1	0	0	0	1	0	0	0	1	27	0	1	0	0	0	0	0
15	green_shoes	0	0	0	10	0	0	2	0	0	0	5	0	2	35	0	0	0	0	1	4
16	green_shorts	0	1	0	0	3	0	0	0	0	0	0	0	0	0	10	0	0	0	0	2
17	red_dress	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98	1	2	0	0
18	red_pants	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	5	44	2	0	0
19	red_shoes	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	9	3	83	0	1
20	white_dress	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	4
21	white_shoes	0	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	1	11	70
23	Accuracy =	78.8%																			

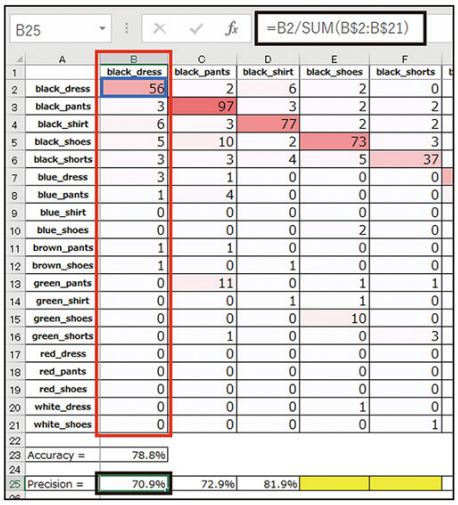
$$\text{Accuracy} = \frac{\text{予測カテゴリと実測カテゴリが一致しているデータ全数}}{\text{全データ数}} \fallingdotseq 78.8\%$$

Accuracy が計算できるということは、Precision や Recall も計算できるのでしょうか。結論としては当然計算できるのですが、多値分類問題の場合は、Confusion Matrix が 3 × 3 以上の行列となるので、**各カテゴリにおける Precision、Recall を計算する**ことになります。

Precision の場合は「**ある予測カテゴリに該当するデータ数のうち、予実のカテゴリが一致している（正解している）データの割合**」と捉えることができます。実際に Excel の 25 行目にて、それぞれの予測カテゴリごとの Precision を計算しています（列が予測カテゴリです）。たとえばセル B25 に関しては、black_dress カテゴリの Precision を計算し、約 70.9% となっていますね。なお、E ～ F 列目の black_shoes、black_shorts カテゴリ部分を空欄にしておいたので、計算してみてください[※16]。

（※ 16）Excel の解答例は「test_result_answer.xlsx」ファイルとして格納しています。

➲ カテゴリごとの Precision も計算できる [図6-4-5]



B25 =B2/SUM(B$2:B$21)

	A	B	C	D	E	F
1		black_dress	black_pants	black_shirt	black_shoes	black_shorts
2	black_dress	56	2	6	2	0
3	black_pants	3	97	3	2	2
4	black_shirt	6	3	77	2	2
5	black_shoes	5	10	2	73	3
6	black_shorts	3	3	4	5	37
7	blue_dress	3	1	0	0	0
8	blue_pants	1	4	0	0	0
9	blue_shirt	0	0	0	0	0
10	blue_shoes	0	0	0	2	0
11	brown_pants	1	1	0	0	0
12	brown_shoes	1	0	1	0	0
13	green_pants	0	11	0	1	1
14	green_shirt	0	0	1	1	0
15	green_shoes	0	0	0	10	0
16	green_shorts	0	1	0	0	3
17	red_dress	0	0	0	0	0
18	red_pants	0	0	0	0	0
19	red_shoes	0	0	0	0	0
20	white_dress	0	0	0	1	0
21	white_shoes	0	0	0	0	1
23	Accuracy =	78.8%				
25	Precision =	70.9%	72.9%	81.9%		

(black_dress の) Precision ＝

$$\frac{\text{予測カテゴリと実測カテゴリがともに black_dress であるデータ数}}{\text{予測カテゴリが black_dress であるデータ全数}}$$

≒ 70.9%

同様に Recall も計算できます。Recall の場合は「**ある実測カテゴリに該当するデータ数のうち、予実のカテゴリが一致している（正解している）データの割合**」と捉えられます。実際に Excel の W 列にて、それぞれの予測カテゴリごとの Recall を計算しています（行が実測カテゴリです）。 たとえばセル W21 に関しては、white_shoes カテゴリの Recall を計算し、約 81.4% となっていますね。こちらも同じく、5 ～ 6 行目の black_shoes、black_shorts カテゴリ部分を空欄にしてあるので、計算してみてください。

➲ カテゴリごとの Recall も計算することができる [図6-4-6]

	A	L	M	N	O	P	Q	R	S	T	U	V	W
1		brown_shoes	green_pants	green_shirt	green_shoes	green_shorts	red_dress	red_pants	red_shoes	white_dress	white_shoes		Recall =
17	red_dress	0	0	0	0	0	98	1	2	0	0		97.0%
18	red_pants	0	0	0	0	0	5	44	2	0	0		80.0%
19	red_shoes	3	0	0	0	0	9	3	83	0	1		83.8%
20	white_dress	0	0	0	0	0	0	0	0	106	4		95.5%
21	white_shoes	0	0	0	0	0	0	0	1	11	70		81.4%

(white_shoes の) Recall ＝

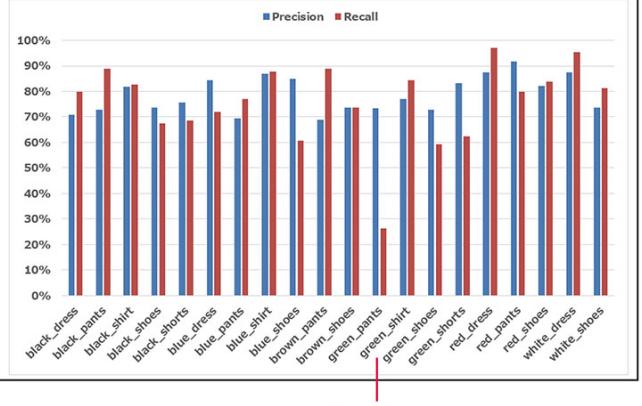
$$\frac{\text{予測カテゴリと実測カテゴリがともに white_shoes であるデータ数}}{\text{実測カテゴリが white_shoes であるデータ全数}}$$

≒ 81.4%

これらの結果を、「精度評価指標」シートにまとめてあります。まとめたとはいっても「Confusion Matrix」シートで計算した Accuracy、Precision、Recall をまとめているだけです。black_shoes、black_shorts カテゴリの Precision, Recall の計算（[図 6-4-7] の黄色部分）ができれば、サマリ表および図が完成します。こうすることで、全体の精度感を確認できます。今回はこれ以上手を施しはしませんが、実務的には精度評価指標を確認し、全体的にモデルの精度を向上させる必要があるか？あるいは一部のカテゴリだけ精度を改善すべきか？といったモデル向上のための課題を見つけ、必要に応じてモデル改善を試みることになります。

➲ Accuracy、Precision、Recall の結果サマリ [図6-4-7]

	A	B	C
1	Accuracy =	78.8%	
2		Precision	Recall
3	black_dress	70.9%	80.0%
4	black_pants	72.9%	89.0%
5	black_shirt	81.9%	82.8%
6	black_shoes	73.7%	67.6%
7	black_shorts	75.5%	68.5%
8	blue_dress	84.5%	72.1%
9	blue_pants	69.4%	77.0%
10	blue_shirt	86.9%	87.7%
11	blue_shoes	84.9%	60.8%
12	brown_pants	69.0%	88.9%
13	brown_shoes	73.8%	73.8%
14	green_pants	73.3%	26.2%
15	green_shirt	77.1%	84.4%
16	green_shoes	72.9%	59.3%
17	green_shorts	83.3%	62.5%
18	red_dress	87.5%	97.0%
19	red_pants	91.7%	80.0%
20	red_shoes	82.2%	83.8%
21	white_dress	87.6%	95.5%
22	white_shoes	73.7%	81.4%



Accuracy やカテゴリごとの Precision/Recall を可視化して、精度を確認できる

ビジネス上のKPIを効果検証する

画像分類モデルの導入で、どの程度のビジネスインパクトが見込めそうかを検証したいところです。今回は、構築した画像分類モデルをフリマサイト（あるいはアプリ）に組み込み、**ユーザーが画像をアップロードしたら、モデルにより予測カテゴリを表示するというシステム連携**となります。そして、その連携によりユーザーの手入力工数が減り、出品手続きが簡素化され、KPI である「出品完了率」の改善を見込みます[※17]。出品完了率が X% 改善

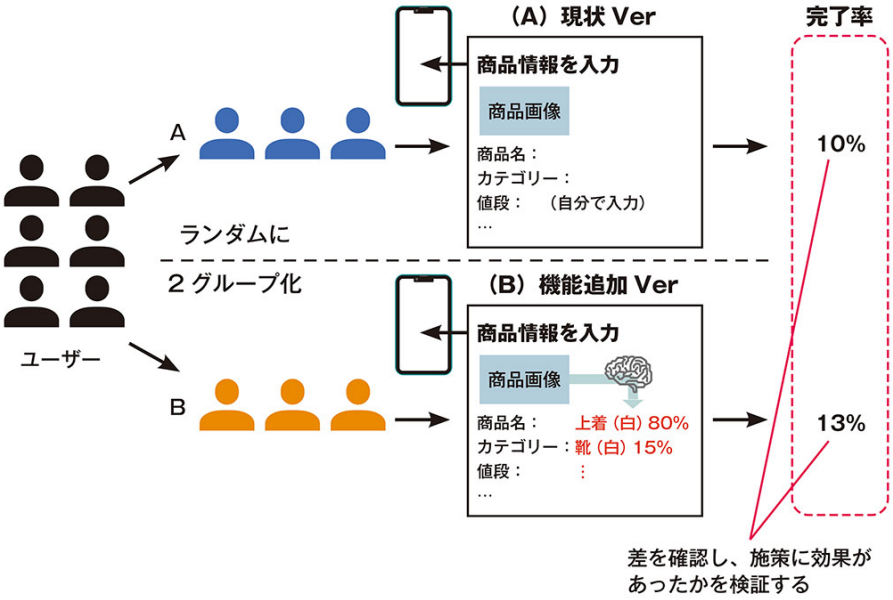
（※ 17）より細かい KPI 設定が可能であれば、画像のアップロードページから次ページへの遷移率、といったより詳細な KPI をモニタリングする形も考えられます。

されることにより、出品数がY%増え、購入数がZ%増えれば、購入手数料が増え、売上効果が見込めます。

ただし今回のようなケースは、実際にモデルをシステム連携してみないと出品完了率の増減がわからず、その出品完了率の増減によりそのほかのKPIが改善するかもわかりません。つまり、シミュレーションによる効果試算が難しいといえます。このような場合は**「ABテスト」**と呼ばれる、打った施策がKPIにどう影響するかを検証する実験手法を実施して効果検証をすることが一般的です。この手法は、**実験対象ユーザーをランダムにA群とB群に振り分け、片方の群にはこれまでと同様の施策、他方の群には新しい施策を打つことで、施策により結果の差を検出する手法**です。今回であれば、A群にはこれまでと同様、B群には商品画像から予測カテゴリを表示する機能を追加した場合とし、実際にユーザーの行動履歴をトラッキングし、KPIを測定します。そして両者で明確な差を検出できれば、今回の施策の効果があったと判断できます。逆にいえば、**実際にシステム連携をしてABテストを実施するハードルがある**という点には注意が必要です。ただし昨今ではABテストを気軽にできるサービスも存在するので、システムの連携さえできてしまえば、ABテストをすること自体は難しくないでしょう。むしろ、**ABテストのように定量的に効果を検証する習慣が重要**といえます。

なお、ABテストによる差の検証は、単純に両者で少しでも差があればOKというわけではなく、本来は**統計学的検定法**を用いて、統計的に考えても、きちんと差があるかどうかをチェックする必要があります。検定の具体的な手順や内容の理解をしようと思うと、統計学の基礎的な部分から押さえないとならないため、本書ではその説明は省略します。ただし検定の内容はわからずとも、ABテストによる差の検証は、**単に少しでも差があればよいわけではなく、統計学的に有意な差が検出されているかどうかを確認してはじめて検証ができたといえる**、という点は意識しておきましょう。

➲ 一般的には、ABテストで効果検証をする [図6-4-8]



Tips　ビジネス適用のイメージを膨らませる

本書では画像解析に関して、まずは基本的な手法である「画像分類」を取り上げましたが、第2章などで紹介したものも含め、物体検出・物体追跡・姿勢推定・画風変換・画像生成など、さまざまな問題に対する技術適用が近年では非常に盛んとなっています。また構造化データに対する適用や、テキスト解析（自動翻訳や文章生成・要約）など、ディープラーニングをさまざまなデータに活用する動きも加速しています。また技術発展に伴って、ビジネスへの実活用の事例もどんどん増えてきており、今後もその流れはどんどん広まっていくでしょう。

実際にそういった新しいアルゴリズムを開発するのは研究者であり、ビジネス適用するための技術応用もデータサイエンティストやエンジニアが実装することになりますが、今回紹介したものを含めて、ディープラーニングを中心とした技術をビジネス適用することに対する抵抗感を払拭したり、こういった技術であれば、自分たちのビジネスに適用できないか？といったイメージを膨らませたりすることは、すべてのビジネスパーソンにとって重要ではないかと感じています。

■ ここで学んだ重要トピック

- 画像分類問題
- ニューラルネットワーク
- Deep Neural Network（DNN）
- Convolutional Neural Network（CNN）
- Pooling（Average Pooling, Max Pooling）
- 畳み込み層（Convolution）
- AB テスト

■ ステップアップにつながるトピック

- 勾配降下法、確率的勾配降下法、AdaGrad、Adam など
- 誤差逆伝播法
- 活性化関数
- Softmax 関数
- Stride、Kernel、Padding など
- Dropout、Batch Normalization
- Attention、Transformer、BERT
- 発展的な画像解析手法

ディープラーニングや画像認識って聞くと、すごく難解なイメージがあったかもしれないけど、学んでみてどうだった？

仕組みがわかってみると、そういうことだったのかと腑に落ちました！アイデア次第でさまざまな応用もできそうです！

Chapter 7

教師なし学習で
ユーザーセグメントを精緻化する

Section

01 ユーザーセグメントを精緻化して施策を出し分けしよう

メールを送る施策のときに、ターゲットごとに内容を変えたじゃないですか？

ええ、ゼロイチの分類問題に落とし込んだ案件のことね。

そうです。今回も同じような案件で、既存客ごとに施策の打ち手を変えたいんです。でも、今回は過去に購入した実績のある人に向けた施策なので、買った買わないのゼロイチでは分類できなそうで……。

なるほどね。それはデータサイエンス的にいえば、「目的関数が存在しないデータを分類する」ということになるわね。そういうときに使えるのが「教師なし学習」よ。

データサイエンス的じゃなく、僕にもわかるように説明してください！

いいわ！ イチから説明するから、ぜひその案件を受注してきてね。授業料は出世払いでいいわよ！

ここで学ぶこと

- ☑ 教師あり学習と教師なし学習の違い
- ☑ k-means 法の基本的な仕組み
- ☑ k-means ほうを使ったユーザーセグメントの切り分け

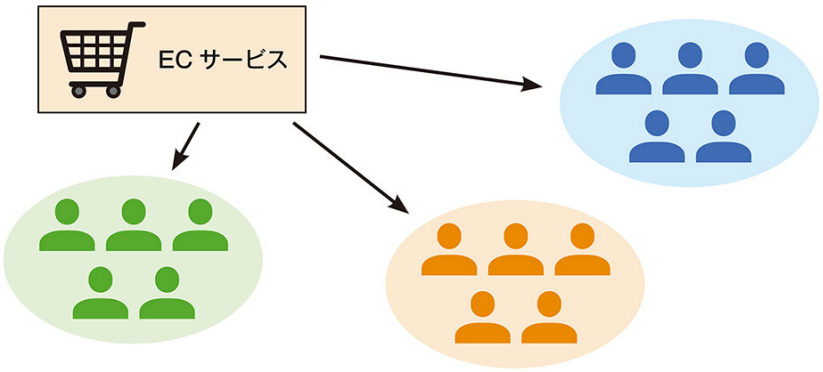
とあるECサイトにおけるマーケティング上の課題を考えてみよう

とある EC サイトの運営会社 E 社のケースを考えてみましょう[※1]。本 EC サイトでは、あるブランドを中心に取り扱っており、これまではユーザー数の獲得に重きをおいてビジネスを展開してきました。それが功を奏しある程度ユーザー数が確保できた現在、次なる成長を見据え、既存ユーザーに対する施策の精度を上げていきたいと考えています。

既存ユーザーへの施策は非常に多岐に渡りますが、1 つの大きい方向性として、これまでの**画一的な施策を見直し、ユーザーセグメントごとに施策を出し分けていこう**という考えがあります。多くのユーザーを抱えているため、さまざまな傾向をもったユーザーが存在しているはずです。

これまではそのような中でも、特に気にせず画一的に施策を打ってきましたが、「こういった傾向を持つユーザーにはこういった施策を」といった形で、傾向の異なるユーザーセグメント（集団）をまずは把握し、そのセグメントごとに施策を出し分けていくこととしました。

➲ 既存ユーザーのセグメントごとに施策を出し分けていきたい［図 7-1-1］



データサイエンスで解くための問題設定

ユーザーごとに施策を分けるためには、傾向の近いユーザーセグメントを把握する必要があります。もちろんユーザーを観察して定性的に判断するなどの考え方もありますが、**今回はユーザーのデータからセグメントを見極め**

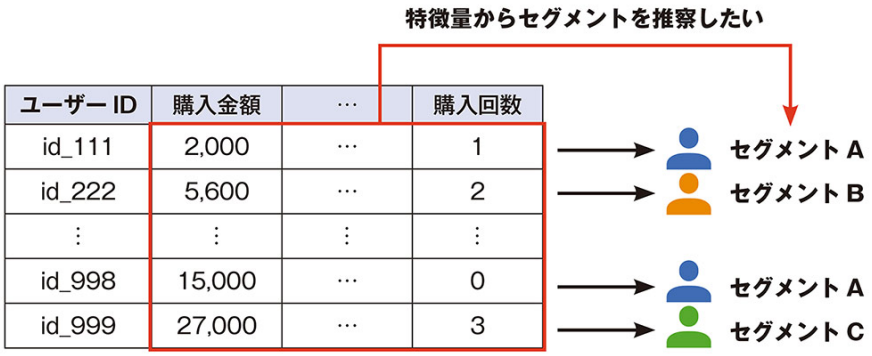
（※ 1）本章もわかりやすい例として EC サービスをもとに考えていきますが、同様の課題を持つ事業やサービスにおいても、もちろん同じような活用が考えられます。

ていきたいと考えています。さまざまなやり方がありますが、「教師なし学習」という方法論を用いて解いていきましょう。

教師なし学習の詳細は次 Section 以降で紹介しますが、これまでの教師あり学習で学んだことと同様に、ユーザーの特徴量を定義する必要があります。ユーザーの特徴量は、いわばユーザーの情報にほかならないので、**どういったユーザーの情報からセグメントを判断するか？**を決めることに等しいといえます。たとえば、ユーザーの年齢情報だけを使って若年層・中年層・シニア層というセグメントに分ける、というのも本質的には特徴量を用いたクラスタリングであると捉えられます。それを発展させて、考慮したい特徴量の数が多くなった際に、単なる 1 軸のセグメント切りではなく、クラスタリングアルゴリズムを使う効果が見込めます。したがって、いったんユーザーの特徴量を定義できれば、後述する教師なし学習のアルゴリズムを使うことで、どのユーザーがどのセグメントに属しているか？を算出できます。

教師なし学習とはどのような考え方か、教師なし学習をどのように利用することでユーザーセグメントを把握できるか？を学んでいきましょう。

➲ ユーザーの情報＝特徴量を定義して、セグメントを把握する［図 7-1-2］



Section

02 教師なし学習の概要

「教師あり学習」と「教師なし学習」の違い

教師なし学習の解説に入る前に、まずはこれまで学んだ教師あり学習との違いを明確にしておきましょう。両者の違いはその名の通りで、「**教師となるデータがあるかないか**」です。教師となるデータというのは、これまで学んだ「目的変数」を指します。

・教師あり学習：教師データである目的変数が存在する
・教師なし学習：教師データである目的変数が存在しない

➲ 教師となるデータ（＝目的変数）があるかどうかが一番の違い［図7-2-1］

教師あり学習の場合

ID	特徴量 A	…	特徴量 N	目的変数
id_001	2,000	…	1	1,000
id_002	5,600	…	2	7,500
⋮	⋮	⋮	⋮	⋮
id_998	15,000	…	0	3,000
id_999	27,000	…	3	4,000

教師なし学習の場合

ID	特徴量 A	…	特徴量 N
id_001	2,000	…	1
id_002	5,600	…	2
⋮	⋮	⋮	⋮
id_998	15,000	…	0
id_999	27,000	…	3

これまでの教師あり学習では、次ページの［図7-2-2］のように、教師となる目的変数を（特徴量から）学習することで、特徴量から目的変数を予測できるようになる、という構図でした。

➲ 教師あり学習は特徴量から目的変数を予測［図 7-2-2］

・“ 販売数 ” という目的変数を学習し、予測する
（需要予測）
・“CV するかどうか ” という目的変数を学習し、予測する
（ユーザーターゲティング）

教師なし学習の概念

もう少し具体的に、教師なし学習のコンセプトを理解しましょう。ユーザーの情報となる特徴量を定義できたら、その特徴量データを教師なし学習アルゴリズムにインプットします。すると、教師なし学習のアルゴリズムは、このデータ群はこのグループ……といった形で、データをいくつかのグループに選別します。

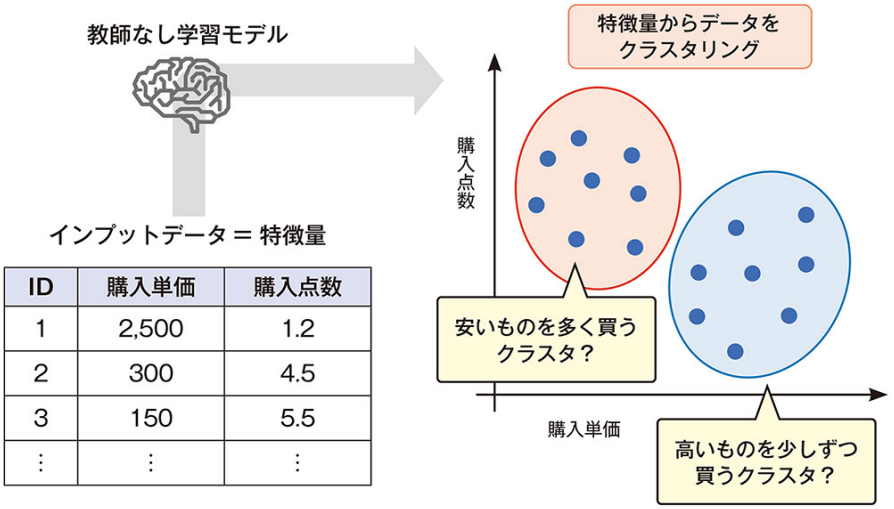
この行為はしばしば「**クラスタリング**」と呼ばれ、クラスタリングによって生成したグループを「**クラスタ**」を呼びます。クラスタもグループもセグメントも同様の意味と思って差し支えありませんが、今後は「クラスタ」という単語で統一します。

クラスタリングによりデータをいくつかのクラスタに分けたら、それぞれのクラスタに関して、クラスタごとの特徴量の傾向を読み解きます。そうすることでどういったクラスタとなっているかを把握でき、クラスタごとの適切な施策を考えられるようになります。

たとえば次ページの［図 7-2-3］のように、購入単価と購入点数でクラスタリングをしたことで、高いものを少しずつ買うようなクラスタや、安いものを多く買うクラスタがいそうであるという読み解き・解釈をするといったイメージです。

どのように教師なし学習がクラスタリングをしているのか、そしてクラスタリング結果を解釈するのか、といった具体的な話は、後ほど紹介します。

➲ データからユーザーをグルーピングする［図7-2-3］



ID	購入単価	購入点数
1	2,500	1.2
2	300	4.5
3	150	5.5
⋮	⋮	⋮

散布図ではダメ？ 高次元になったときを考える

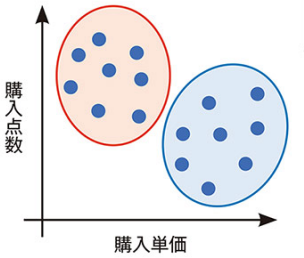
さて、先ほどの［図 7-2-3］を見ると「わざわざクラスタリングをせずとも、特徴量データの散布図を見ればいいのではないか？」と思ったかもしれません。たしかに散布図でも事足りそうですが、たとえば加味したい特徴量の数が 10 個あったらどうでしょうか。その場合、10 個の特徴量データの傾向を同時に可視化することは難しいでしょう。

このように、特徴量数が多い、つまり対象とするデータの列数が多い状態をよく「**高次元**」といいますが、高次元データを取り扱う場合は、可視化による把握は難しいといえます。もちろん、同時に可視化せずとも、特徴量の列を１つ１つ取り出して集計・可視化することは可能ですが、あくまでそれは１つの特徴量だけを切り取って見ているに過ぎません。

したがって、**複数の特徴量を「同時に」加味して、データのクラスタを把握したい際には、教師なし学習によるクラスタリングの効果を発揮できる**でしょう。

➲ 高次元データの場合、可視化による把握が難しい［図7-2-4］

ID	購入単価	購入点数
1	2,500	1.2
2	300	4.5
3	150	5.5
⋮	⋮	⋮



ID	購入単価	購入点数	購入回数	年齢	…
1	2,500	1.2	10	50	…
2	300	4.5	2	25	…
3	150	5.5	5	30	…
⋮	⋮	⋮	⋮	⋮	⋮

?

特徴量の列が多い＝高次元であると、可視化により把握することが難しい

多くの特徴量を同時に加味して、データのクラスタを把握したいときに「教師なし学習」を利用できる！

Tips　高次元データを低次元に圧縮する

実は、高次元なデータを、列数が少ない＝低次元な状態へ圧縮する「**次元圧縮**」といった手法もしばしば用いられます。低次元に圧縮するとは、たとえば10個（10列）の特徴量を、その情報量をできるだけ保ったまま2個（2列）にしてしまう、といったイメージです。低次元に圧縮することで、データを可視化しやすくなったり、軽量になるため取り扱いやすくなったり、といった利点があります。

ただしその一方で、次元を圧縮しているため、元のデータより情報量が落ちてしまったり、あるいは圧縮された各列の意味を判断するのが難しくなったりといったデメリットも存在します。したがって適切な使いどころは難しいのですが、そのような方法論があるということは知っておいて損はないでしょう。

基本的には、次元圧縮などを用いなければ高次元データを直接的に可視化することは難しいと考えて差し支えありません。

Section

03 教師なし学習の基本手法「k-means 法」

教師なし学習のアルゴリズム

本 Section では、教師なし学習がどのようにクラスタリングをしているのかというロジックを理解しましょう。教師あり学習と同様に、教師なし学習にもさまざまなアルゴリズムが存在します。

➲ 教師なし学習のアルゴリズムの一例［図 7-3-1］

- k-means 法、k-means++ 法
- x-means 法
- スペクトラルクラスタリング
- DBSCAN
- 混合ガウスモデル
- 階層型クラスタリング

今回は皆さんが、どのようにクラスタリングが行われるのか？というイメージが一番つきやすいように、ロジックが比較的シンプルで、かつ基本的で有名なアルゴリズムである「**k-means 法**」として取り上げます。

k-means法とは？

まず k-means 法の「k」に重要な意味があり、**これはデータをいくつのクラスタに分割したいかというクラスタ数 k** を意味します。k-means 法を含めいくつかのクラスタリング手法では、**クラスタ数を指定する必要**があります。これは少々面倒なことのように思えますが、そもそもどのようなユーザー

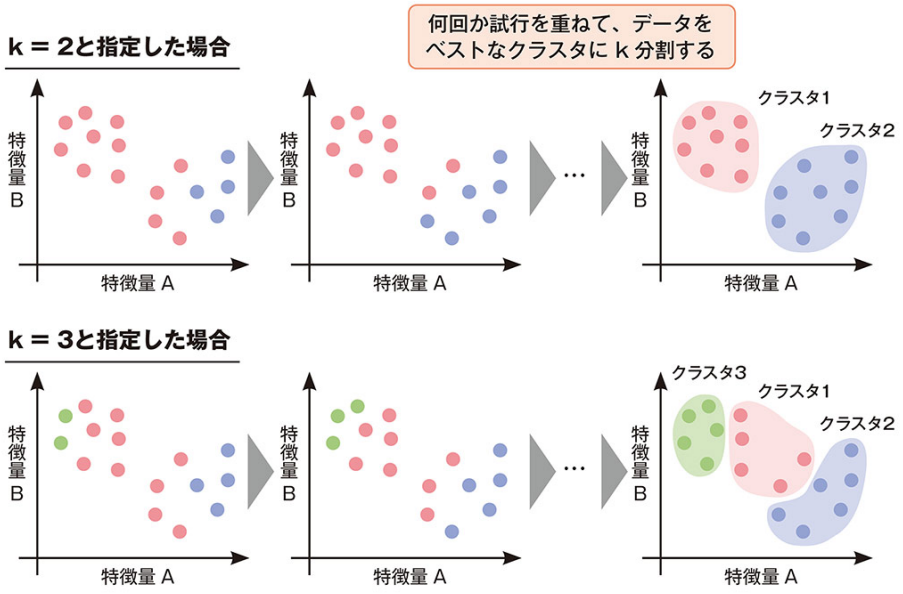
セグメントが存在するかがわかっていないので、致し方ない部分もあります。

また、もしそうなるとｋをどう設定すればよいのか？と思うでしょう。このｋの決め方は実務的にもとても難しいですが、いくつかのパターンで試してみて、一番解釈しやすいクラスタ数ｋと定めることが一般的です。このｋの決め方に関しては、本章後半の実データによる演習のSectionで、もう少し詳細に取り上げます。

さて、仮にｋを２や３と指定したとしましょう。すると、［図7-3-2］の一番左の図のように、アルゴリズムがデータをｋ分割します。**最初はどのように分割するのがベストかわからないため、ランダムに分割**します。

最初の段階はいわば大ざっぱに分割しているので、よりよいクラスタに分割できるはずです。そこで、数学的によりよい方向に修正できるように計算し直す形で、**何回か試行を重ねていくことで、ベストなクラスタにｋ分割**できます。

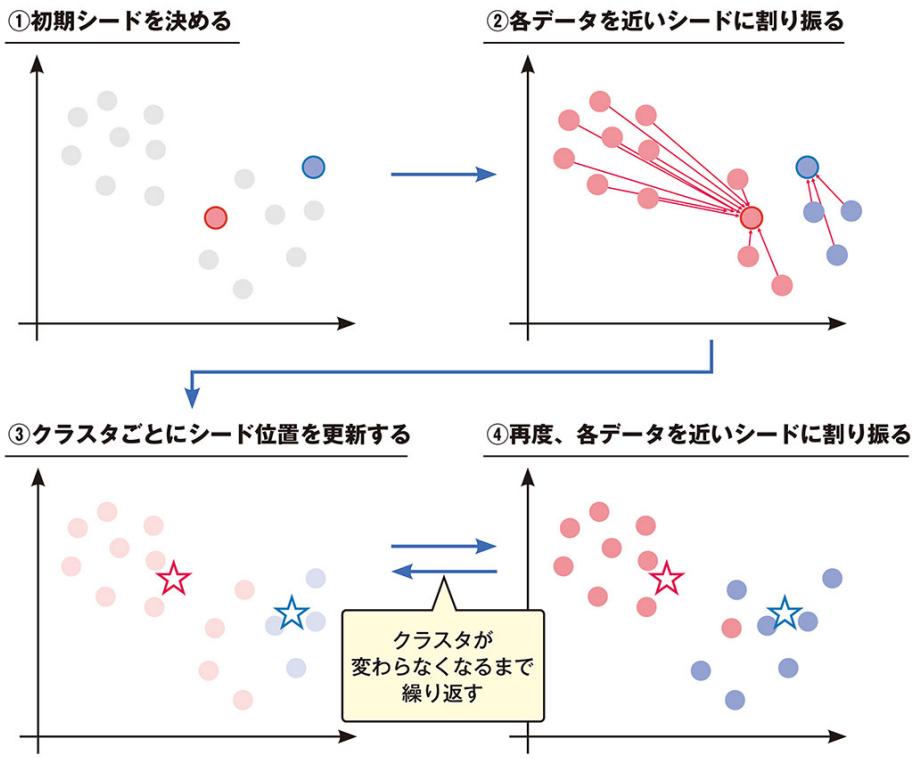
➲ クラスタ数ｋを決めることで、データをベストなクラスタにｋ分割する
［図7-3-2］



k-means法のアルゴリズムの詳細

k-means 法でどのように試行を重ねているかというアルゴリズムの詳細を紹介します。k-means 法では、［図 7-3-3］のようなステップでクラスタリングを行います。

➲ k-means 法のクラスタリングアルゴリズムのステップ［図7-3-3］



最初は何も情報がないので、Step1 としてデータから**ランダムに k 個を選択し、「初期シード」**（①）とします。

Step2 として、初期シード以外のデータに関して、それぞれの**データがどのシードに近いかを割り振り**ます（②）。基本的には距離が近いシードに所属させることになります。これで一度目のクラスタが生成されたことになります。

よりクラスタの精度を上げるために、Step3 として、最初に設定したシー

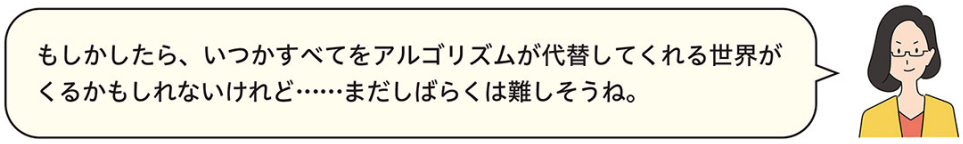
ドの位置を更新します。その方法としては、**クラスタごとの中心の位置（重心）をシードの位置として更新**（③）することが一般的です。

するとシード位置が変わるので、Step2 と同様に、再度それぞれのデータに関して近しいシードに割り振ります（④）。シード位置が変わっているので、いくつかのデータは割り振られるシード（クラスタ）が変わっているはずです。これで、クラスタが更新されたことになります。

これにより、クラスタごとに所属するデータが変わったので、中心の位置がずれているはずです。したがって、Step3 のシード位置更新、Step4 のデータのシードの割り振り、を繰り返します。最終的にどこかのタイミングで、シード位置が動かずに、クラスタごとに割り振られるデータが固定されます。そのように**収束したら、アルゴリズムによる処理を終了**させます。これで「学習」が完了したことになります。

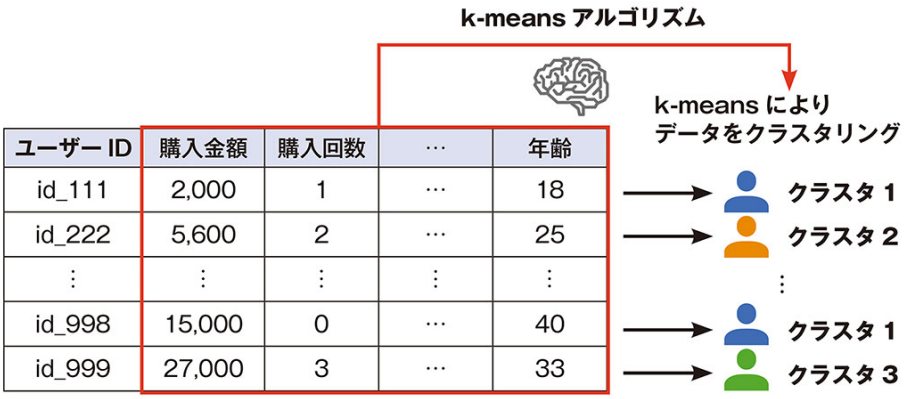
特徴量を定義してクラスタリングする

教師あり学習でも同様ですが、k-means 法だけではなくすべての教師なし学習アルゴリズムは、**与えられたデータに関して、最適なクラスタを発見するための道具**です。裏返すと、教師なし学習アルゴリズムに対して、私たちが適切にデータを与える必要があるということです。ここでいうデータは特徴量です[※2]。したがって、**どういった特徴量がユーザーのセグメント（クラスタ）に寄与していそうか？**という観点で特徴量を洗い出し、データから適切に特徴量を生成する必要があります。この行為は教師なし学習アルゴリズムはやってくれないので、私たちが**「どういったユーザークラスタが存在しそうなのか？」という仮説を立てながら、適切に教師なし学習アルゴリズムを使いこなすことが必要**となります。



（※2）細かいことをいえば、どのようなユーザー集団とするか？どの期間のデータとするか？といった論点も存在します。

➲ 定義した情報＝特徴量のデータにもとづいてクラスタリングする［図7-3-4］



ユーザー ID	購入金額	購入回数	…	年齢
id_111	2,000	1	…	18
id_222	5,600	2	…	25
⋮	⋮	⋮	⋮	⋮
id_998	15,000	0	…	40
id_999	27,000	3	…	33

k-means法の注意点

これまで紹介した k-means 法には、押さえておくべき注意点がいくつか存在します。細かい部分もあるため、ここでは最低限押さえておきたいポイントのみ紹介します。

まず大きな注意点は、先述しましたが、**クラスタ数 k を事前に決めないとアルゴリズムによる学習ができない**という点です。したがって実務的には、いくつかのクラスタ数 k で試してみて解釈しながら適切な k を選択していくことになります（後述の演習で取り上げます）。

もう 1 つは、k-means 法のアルゴリズムロジック（［図 7-3-3］）の Step1 における、データからランダムに k 個を選択する「初期シード」に関してです。ここではランダムに選択するので、**初期値に依存して学習結果が変わってしまう可能性がある**ということです。このような現象を「**初期値依存性**」とよくいいますが、アルゴリズムではどうしてもランダムに何かをするという必要性を排除できない場面もあり、よく起きてしまいます。

Section

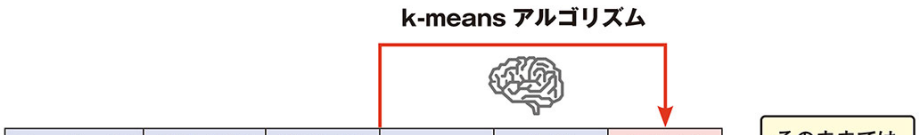
04 クラスタリング結果の解釈

クラスタリング結果を確認する

教師「あり」学習では、学習・予測した値を何かしらの業務やサービスの中に組み込むことで、業務効率化やサービスの高度化を実現していました。したがって、機械学習における教師あり学習では、その精度自体が担保されているかどうかをしっかりと評価することに重点が置かれていました。

一方で教師「なし」学習の場合は、学習によるクラスタリングができたら、その結果を解釈する必要があります。教師なし学習の場合も、学習できたら、[図7-4-1]のように**ユーザーごとにクラスタIDが振られます**[※3]。これにより、どのユーザーがどのクラスタに割り振られているかは判断できます。しかし、このままではただ採番がされているだけで解釈できません。このクラスタリング結果の状態から、解釈をする必要があります。

➲ クラスタリングにより、ユーザーごとにクラスタIDが振られる［図7-4-1］



ユーザーID	購入金額	購入回数	…	年齢	クラスタ
id_111	2,000	1	…	18	1
id_222	5,600	2	…	25	3
⋮	⋮	⋮	⋮	⋮	⋮
id_998	15,000	0	…	40	2
id_999	27,000	3	…	33	1

そのままでは解釈できない

（※3）なお、クラスタIDの数字自体はただ順番に振られた連番なため、意味はありません。

クラスタごとの特徴量の傾向を把握する

クラスタリング結果を解釈する方法はさまざまですが、ここでは比較的わかりやすく一般的な考え方を紹介します。1つの方法としては、**各クラスタに属しているデータの、特徴量それぞれの統計量、たとえば平均値などを集計**してみるとわかりやすくなります。[図7-4-2]の表のようなイメージです。このように集計することで、ただデータごとにクラスタIDが振られていた状態よりも、結果が見やすくなります。

そして、**集計結果から、その傾向を読み取り**ます。たとえばクラスタ1であれば、そのほかのクラスタと比較して「年齢層は比較的若めであり、購入金額はあまり高くなく、また購入回数も少ない傾向にありそうだ」といったことが読み取れそうです。

➲ クラスタごとの特徴量の統計量の傾向を見て解釈を試みる［図7-4-2］

クラスタ	購入金額の平均値	購入回数の平均値	…	年齢の平均値
1	2,000	1.2	…	22.5
2	12,000	2.0	…	67.5
3	5,000	10.3	…	35.3

比較的若年層で、購入金額も購入回数も少ない？

比較的高年齢層で、購入回数は少ないが、1回あたりの購入金額は高そう？

比較的中堅年齢層で、購入回数が多そう？

クラスタごとに、特徴量の統計量などを確認することで、クラスタの解釈が簡単になる

結果が読み取れれば、（あくまで例ですが）クラスタ1に対しては、若年層における購入体験の不足が課題の仮説としては考えられ、若年層をターゲットとして、購入金額や購入回数を増やしてもらえるような施策を考えることができそうです。

Section

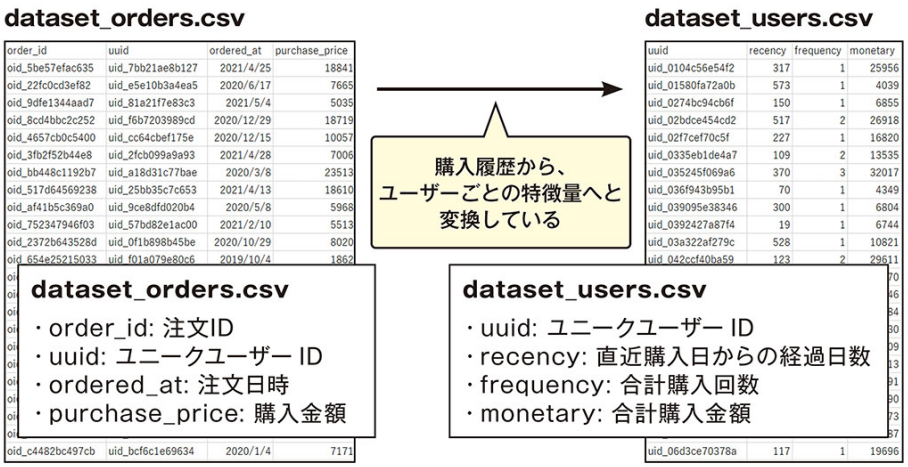
05 実践：EC サイトの購入履歴データを活用しよう

練習用ファイル：chap07_user_clustering / dataset_orders.csv、dataset_users.csv

実践 データの確認

ここでは本章最後の Section として、実践演習を通じて、教師なし学習のビジネス適用のイメージを深めていきましょう。冒頭で述べたように、今回は運営している EC サイトの**既存ユーザーに対して、ユーザークラスタごとに施策を出し分けていくこと**を目指します。今回使用する元データは**購入履歴データ**となり、「dataset_orders.csv」に格納されています（データはすべて「chap07_user_clustering」フォルダに格納しています）。このデータは、**注文 ID ごとユーザー ID ごとの注文日時・購入金額**という、一般的な購入履歴データです※4。しかし今回は教師なし学習アルゴリズムを利用してユーザークラスタを生成する必要があります。そこで、ユーザーごとの特徴量を生成し、データ「dataset_users.csv」のように持ち直します（やり方は次ページで解説します）。

➲ サイトでの購入履歴とユーザーごとの特徴量データを使用する［図7-5-1］



（※4）本来であれば、どういった商品を購入したか、そもそもどのようなユーザー属性か、といったさまざまなデータも存在するはずであり、そのようなデータも活用したいところですが、データ構造をシンプルにしてわかりやすく説明することを主目的としたいため、今回はそのようなデータ属性は省略します。

練習用ファイル：chap07_user_clustering / dataset_orders.csv

実践 購入履歴から特徴量を定義する

購入履歴データから特徴量を作り込む方法にはさまざまなアプローチが考えられますが、今回はよく使われる分析フレームワークである「**RFM 分析**」を利用してみましょう。RFM 分析とは、**「Recency」（最新購入日からの経過日数）**、**「Frequency」（購入頻度）**、**「Monetary」（購入金額）**の 3 つの指標でユーザーを分類する方法です。これらの指標は、今回の購入履歴データから生成することができそうです。なお第 3 章でも紹介しましたが、この特徴量の生成を Feature Engineering と呼びました。

➲ RFM 分析の指標［図7-5-2］

- Recency：（いつでもよいのですが、本日を 2021/6/1 とし、）ユーザーごとに、最新購入日から 2021/6/1 までの日数を計算する
- Frequency：ユーザーごとに、購入した回数を足す
- Monetary：ユーザーごとに、購入金額の合計を計算する

➲ 購入履歴データからユーザーごとの特徴量データに持ち直す［図7-5-3］

購入履歴

注文 ID	UUID	購入日時	購入金額
001	111	21/4/5	1,000
002	111	21/4/30	500
003	111	21/5/15	2,500
⋮	⋮	⋮	⋮
998	199	21/5/20	1,000
999	199	21/5/31	1,300

ユーザーごとの特徴量データ

UUID	Recency	Frequency	Monetary
111	17	3	4,000
112	⋮	⋮	⋮
⋮	⋮	⋮	⋮
199	1	2	2,300

Feature Engineering

購入履歴データから、ユーザーごとの特徴量を計算することで、教師なし学習のデータを得ることができる

上記の定義をもとに計算した結果を「dataset_users.csv」に格納しています。したがってこのデータは、**ユーザー ID（uuid）が一意になっている（uuid がデータセットで 1 行しか存在しない）データ**となっています。

今回は、この dataset_users.csv のデータをそのまま教師なし学習アルゴリズムに放り込み、その結果を解釈してみましょう。

練習用ファイル：chap07_user_clustering / users.xlsx

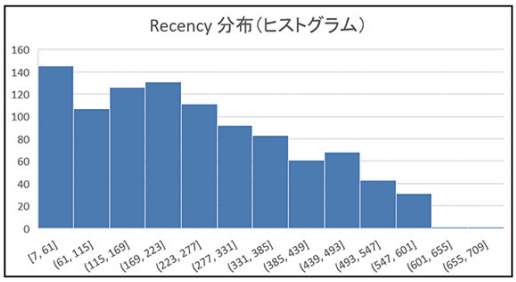
実践 特徴量の傾向を確認する

その前に、簡単ではありますが、今回利用するデータの傾向を理解しておきましょう。特徴量の傾向を可視化した結果を「users.xlsx」に格納してあります。

元データの状態から、Recency と Monetary はヒストグラム（データの分布）を、Frequnecy はピボットテーブルにより値ごとのデータ数を可視化しています。そこからは、［図 7-5-4］右下のような情報や傾向が読み取れそうでしょうか。

➲ RFM それぞれの特徴量の傾向を可視化する［図 7-5-4］

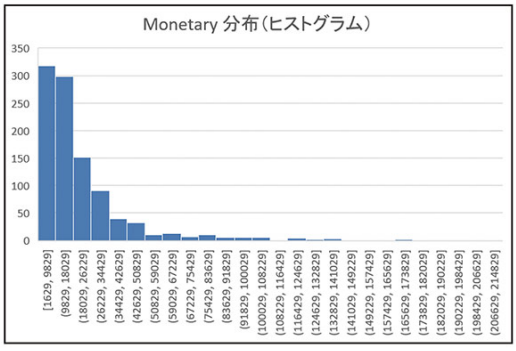
Recency の傾向



Frequency の傾向

行ラベル	個数 / frequency
1	631
2	338
3	23
4	4
5	2
6	2
総計	1000

Monetary の傾向



Recency

・7 ～ 600日間に収まっており、
・比較的多くのユーザーが1年以内には1回購入している

Frequency

・1 ～6回の購入回数に収まっており、
・多くのユーザーは1~2回の購入

Monetary

・右裾が長い分布となっており、
・多くのユーザーは30,000円ほどの購入金額に収まっている

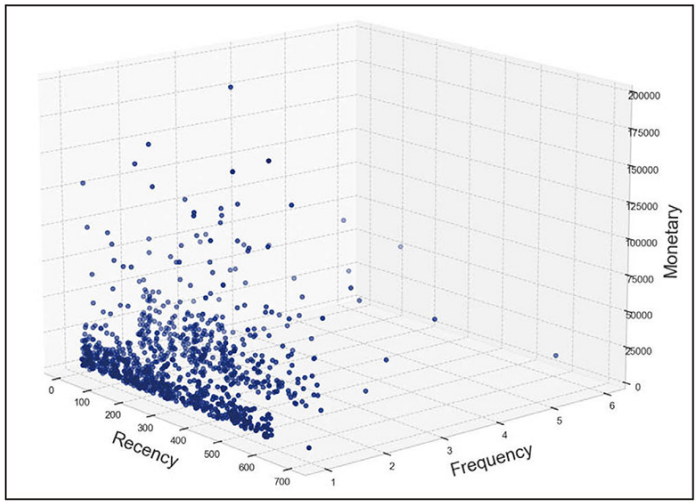
このように、データを直接観察したり、集計・可視化したりしながらデータを理解する営みは**「探索的データ解析」（Exploratory Data Analysis、EDA）**ともいわれます。EDA を通じて、データに外れ値や異常値がないか？興味深い傾向がないか？といったことを調べることで、より精緻にモデル構築をすることが期待できます。実務的には、元データ（購入履歴）を含めて、より詳細に EDA を行うことが求められますが、ここでは紙幅の関係上、この程度としておきます。大きな外れ値やおかしな傾向もなさそうなため、このまま進めていくこととしましょう。

練習用ファイル：chap07_user_clustering / users_result.xlsx

実践 k-meansでクラスタリングした結果を確認する

さて、この R、F、M の特徴量をもったユーザーは、どういったクラスタに分けられそうか、クラスタリングにより確認してみましょう。ちなみに R、F、M の 3 つの特徴量なので、わざわざクラスタリングせずとも、3 次元空間の散布図で何か確かめられるでしょうか。実際に散布図を［図 7-5-5］に示します（Excel での可視化が難しいため、私が別途作成しました）。

➲ R、F、M の 3 つの特徴量空間上にユーザーをマッピング［図7-5-5］



3 次元で可視化しても、わかりにくい

たしかに、3次元空間上に表現できましたが、正直な感想としては、わかりにくいと感じる方が多いのではないでしょうか。そもそも、データを3次元に表現することは物理的に可能ですが、視覚的にわかりにくいことが多く、**可視化の方法として3次元は推奨されていません**[※5]。かつ、仮にある程度見やすかったとしても、そこからどういったクラスタが存在しているかを定量的に判断することは難しいでしょう。そこで、やはり今回学んだ教師なし学習によるクラスタリングが効果を発揮しそうです。

実際に通常版のExcelで教師なし学習を実装するのは難しいため、これまでと同様に、私がPythonによりk-means法を実装した結果を観察してみましょう。その結果を「users_result.xlsx」に格納してあります。結果を見ると、k-meansのクラスタ数であるkが2、3、4の3パターンの結果があることが見てとれます。次節では、このk=2、3、4それぞれの場合に関して、クラスタごとにどういった解釈ができるかを考えてみましょう。

➲ k=2、3、4におけるk-means法によるクラスタリング結果［図7-5-6］

uuid	recency	frequency	monetary	labels_in_cluster_2	labels_in_cluster_3	labels_in_cluster_4
uid_0104c56e54f2	317	1	25956	2	3	4
uid_01580fa72a0b	573	1	4039	2	3	4
uid_0274bc94cb6f	150	1	6855	2	2	3
uid_02bdce454cd2	517	2	26918	1	3	4
uid_02f7cef70c5f	227	1	16820	2	2	3
uid_0335eb1de4a7	109	2	13535	1	1	2
uid_035245f069a6	370	3	32017	1	1	2
uid_036f943b95b1	70	1	4349	2	2	3
uid_039095e38346	300	1	6804	2	2	3
uid_0392427a87f4	19	1	6744	2	2	3
uid_03a322af279c	528	1	10821	2	3	4
uid_042ccf40ba59	123	2	29611	1	1	2
uid_049b5e94eb93	168	1	10570	2	2	3
uid_04e863d3057e	194	1	18446	2	2	3
uid_04fca0044b68	479	1	29884	2	3	4
uid_050ee6d282d5	66	2	12430	1	1	2
uid_053e3c8b2fcd	322	2	129409	1	1	1
uid_05a74e4454a1	413	1	8513	2	3	4
uid_05ea60bd5940	43	1	11491	2	2	3
uid_0630192656ab	513	1	11590	2	3	4
uid_06638fcb2673	124	1	4573	2	2	3
uid_06768aea2b39	99	2	26987	1	1	2
uid_06d3ce70378a	117	1	19696	2	2	3
uid_06e1bd9cf396	466	1	8764	2	3	4
uid_06e921088191	370	2	9798	1	3	2
uid_0742355080fe	238	2	62037	1	1	2

クラスタ数 k=2 の場合

クラスタ数 k=3 の場合

クラスタ数 k=4 の場合

（※5）場合によっては3次元の表現がわかりやすいケースも存在します。

なお、理論的には、この 3 パターンだけではなく、k=2、3、4、5……とすべてのパターンを考える必要がありますが、さすがにそれは難しいですね。そこで、実務的には、**数学的に問題ないであろうクラスタ数に絞り込む**ことが一般的です。その説明は数学的な内容になるので省略しますが、「エルボーメソッド」や「シルエットプロット」といった方法論が存在します。ただし、そのような方法論でも、きれいにクラスタ数をここからここまでと絞り切ることは難しいのが現実です。そこで、そのような**理論的な方法論である程度見込みをつけつつも、最終的には指定したクラスタ数ごとの解釈をしたうえで、ある程度ビジネスに落とし込めそうな結果を取り上げる**、というプロセスを経ることが多いです。

かつ、今回のような施策検討の場合、いくら理論的にはクラスタ数 k=20 がベストであるとわかったとしても、いきなり施策を 20 個考えて、それらすべてを出し分けて実行するのは厳しいでしょう。つまり、**実務的に遂行可能かどうかというフィージビリティの観点も考慮に入れる必要**があります。

今回は、エルボーメソッドといった理論的な方法論によるチェックはある程度精査したうえで、かつはじめての施策の出し分けということもあり、あまり多すぎるクラスタ数は現実的ではないと総合的に判断し、クラスタ数 k が 2、3、4 の 3 つのパターンに絞って考えてみることとします。

練習用ファイル：chap07_user_clustering / users_result.xlsx

実践 クラスタごとの特徴量の傾向を解釈する

まずは k=2 におけるクラスタリング結果の傾向を見ていきます。「result」シートにある結果だけではわかりにくいので、k=2 の場合の結果を集計・可視化したものを「k=2」シートに記載しています。

“labels_in_cluster_2” 列を軸として、クラスタ ID ごとの

・データ（ユーザー）の個数
・R (Recency) / F (Frequency) / M (Monetary) の平均値

を集計した**ピボットテーブル**を左上に、その下には、R、F、M ごとにクラ

スタ ID それぞれの**ヒストグラム（度数分布）**も描画しています（当該ヒストグラムは Excel での描画が難しかったため、私が別途作成した図を添付しています）。

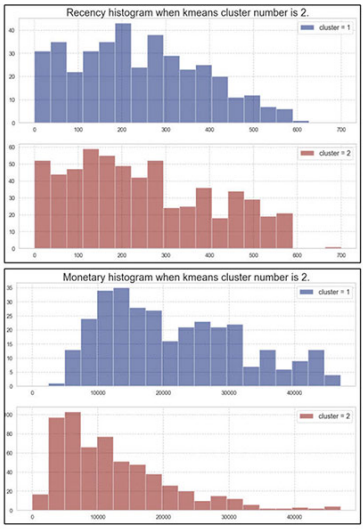
ピボットテーブルの統計量を中心に確認しつつ、ヒストグラムでデータの傾向の詳細をチェックする、といった見方がよいでしょう。これらの傾向を見ると、［図 7-5-7］の右下に記載したようなことがいえそうです。

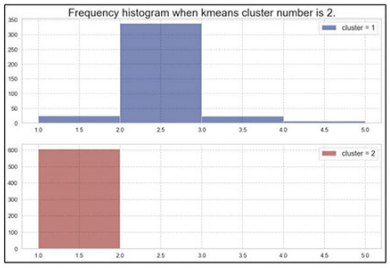
➲ k=2 におけるクラスタリングの可視化（統計量とヒストグラム）［図 7-5-7］

k = 2

行ラベル	個数 / uuid	平均 / recency	平均 / frequency	平均 / monetary
1	393	236.30	2.05	34194.56
2	607	246.29	1.00	12979.58
総計	1000	242.365	1.414	21317.066

クラスタ数 k=2 の場合の
クラスタごとの傾向





1. 全体の1/3強を占め、購入回数が多く、購入金額が高めのユーザー群？
2. 全体の2/3弱を占め、購入回数が少なく、購入金額が低めのユーザー群？

つまり、全体の 1/3 強のユーザー（クラスタ ID=1）は、比較的**ロイヤルティ（顧客からの信頼度）が高く**、残りのユーザー（クラスタ ID=2）はそうではない、といった傾向が見えてきそうです。

ただし、これでは少々ユーザーセグメントへの解像度が粗そうです。もう少し深堀りしてみるために、クラスタ数を増やした結果も確認しましょう。

同様に「k=3」シートに結果を格納しています。結果の見方に関しては、k=2 のときと同様に、ピボットテーブルとヒストグラムの可視化を確認します。［図 7-5-8］の右下に記載しているような傾向が見てとれそうでしょう。

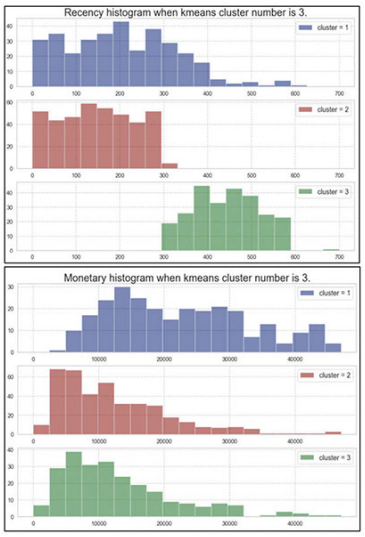
k=2 の際よりも細分化され、先ほど同様ロイヤルティの高そうなクラスタ（クラスタ ID=1）と、あまり高くないクラスタ（クラスタ ID=2）に加えて、新たなクラスタ ID=3 は、Recency の値が相対的にとても大きいです。これはつまり、**直近購入日からの経過日数が長くなっているため、しばらく購入から遠ざかっているのではないか？**と考えられます。

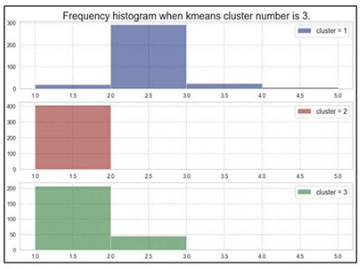
➲ k=3 におけるクラスタリングの可視化（統計量とヒストグラム）［図 7-5-8］

k = 3

行ラベル	個数 / uuid	平均 / recency	平均 / frequency	平均 / monetary
1	342	203.85	2.08	35986.82
2	405	148.39	1.00	12778.27
3	253	444.87	1.18	15155.63
総計	**1000**	**242.365**	**1.414**	**21317.066**

クラスタ数 k=3 の場合の
クラスタごとの傾向





1. **購入回数が多く、購入金額も高いユーザー群？**
2. **購入回数や購入金額に強い傾向はないが、比較的直近に購入してくれているユーザー群？**
3. **しばらく購入から遠ざかっているユーザー群？**

これは、Recency のヒストグラムを表している左上の図の、クラスタ ID=3 の緑色の分布だけ右側にずれていることを見ても明らかです。

最後に、もう一歩詳細に踏み込む形で、クラスタ数 k=4 の場合も見ておきましょう。k=2、3 のときと同様に、ピボットテーブルとヒストグラムの可視化を確認します。k=4 に関しては、ピボットテーブルの集計を空欄（黄色部分）にしておいたので、k=2、3 と同様に集計してみましょう 。

その集計結果を、これまでと同じく［図 7-5-9］右下に記載しているように傾向を読み取ります。クラスタ ID=2、3、4 に関しては、先ほどの k=3 の場合のクラスタと同様の傾向を持っていると推察できます。一方でクラス

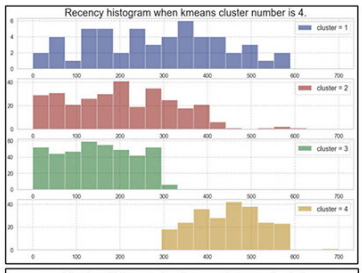
タ ID=1 に関してはユーザー数が 53 と少数ですが、**Monetary（合計購入金額）の平均値が 10 万円近くと非常に高い値**を取っています。Recency や Frequency はそこまで突出した値は取っていないため、**購入頻度はそこまで突出していないが、1 回の買い物における購入金額が非常に高いユーザーが少数ながら存在する**、と推察できそうです。

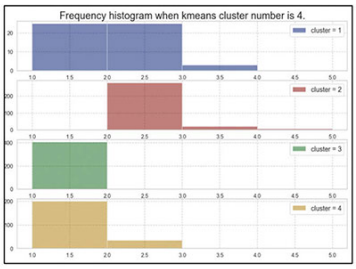
➲ k=4 におけるクラスタリングの可視化（統計量とヒストグラム）［図 7-5-9］

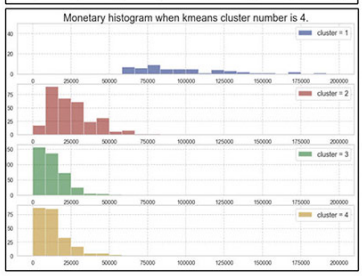
k = 4

行ラベル	個数 / uuid	平均 / recency	平均 / frequency	平均 / monetary
1	53	280.23	1.58	99872.09
2	307	202.24	2.14	24839.40
3	406	148.75	1.00	12821.70
4	234	448.85	1.15	13643.32
総計	1000	242.365	1.414	21317.066

クラスタ数 k=4 の場合の
クラスタごとの傾向







1. 少数だが、**非常に購入金額が高い**ユーザー群？
2. **購入回数が多く**、購入金額も中程度に高いユーザー群？
3. 比較的直近に**購入してくれている**ユーザー群？
4. **しばらく購入から遠ざかっている**ユーザー群？

このように、クラスタリングの結果をうまく集計・可視化することで、その傾向を読み解くことができます。

クラスタに応じた施策を考案する

最後に、解釈できたクラスタに関して、ビジネス的にどういった施策の方向を考えるべきかを検討しましょう。k=2、3、4 といくつかのパターンで考えましたが、クラスタ数が多い k=4 の場合を考えればすべてを充足できそ

うなので、先述した４つのクラスタごとに対する施策を検討しましょう。

それぞれのクラスタの傾向を改めて振り返りつつ、ありえそうな施策を考えてみます。念のためですが、あくまで１つの案であり、実務的にはさまざまな制約や情報にもとづいて柔軟に検討すべきであることに留意してください。また、**今回の特徴量は R、F、M しか利用できておらず、本来であればどういった商品を買っているか？ どういったユーザー属性か？ といった情報（データ）も使用し、特徴量に組み込むことができれば、より精度の高い施策も検討できる**でしょう。あくまで RFM のみの傾向から読み取れる、という制約があることにも留意しておきましょう。

➲ 分析結果に基づく施策案［図7-5-10］

1. 少数だが、非常に購入金額が高いユーザー群（53名）
購入頻度はそこまで突出していないが、1回の買い物における購入金額が非常に高いユーザーです。このようなユーザーは売上への貢献度が高いことが多く、またロイヤルティも高い可能性があります。ユーザー数も少ないことから、他ユーザーへの配慮をしつつも、優待的なクーポンやイベント案内などが効果的かもしれません。
施策上可能であれば、対面によるイベントへの招待もありえますし、あるいは比較的高額な商品が買える層であると仮説を立てて、高額商品の告知やクーポン配布、といった施策へ落とし込めそうです。

2. 購入回数が多く、購入金額も中程度に高いユーザー群（307名）
1ほどではないにしろ、比較的ロイヤルティが高いユーザーであることが予想でき、1回あたりの購入金額にもある程度余裕があることも期待できそうです。そこで、過去の購入商品からオススメのもう1商品を紹介することで、売上向上を目指せるかもしれません。

3. 比較的直近に購入してくれているユーザー群（406名）
Recencyの値が小さい＝比較的直近に購入しているので、購入に対する記憶が新しいユーザーである可能性があります。「鉄は熱いうちに打て」ではありませんが、再度購入してくれる確率が高いことも考えられ、XX日以内にサイトへ再来訪でYY%オフといったクーポン配布などが施策としては効果的かもしれません。

4. しばらく購入から遠ざかっているユーザー群（234名）
このようなユーザーは場合によっては「離反顧客」とされ、施策を打つ対象から外してしまうという可能性もあるかもしれません。しかし、ユーザーを取り戻すための施策も考えられます。たとえば、3と同じクーポン配布にはなりますが、離脱可能性が高くそもそも再来訪してくれる母数が少ない可能性があることを鑑みて、クーポンによる割引率を高めに設定しておき、少しでも可能性のあるユーザーの再来訪確率を上げる、といった施策が考えられます。

これらの施策案は、あくまで一例に過ぎませんが、施策を**画一的に打つよりは、適切にユーザーの傾向を定量的に把握してプロモーションを打つことが期待できます**。

■ ここで学んだ重要トピック

- 教師なし学習
- k-means 法
- クラスタリング結果の解釈

■ ステップアップにつながるトピック

- 初期値依存性の解消、k-means++ 法など
- エルボーメソッド、シルエットプロットなど
- 階層型クラスタリング、スペクトラルクラスタリング
- 次元圧縮（PCA、SVD、t-SNE など）

もちろん、RFM 分析自体にも、また教師なし学習によるクラスタリングも万能ではないので、それらに頼りすぎないことは注意が必要ね。

どういったユーザーが存在するのか？どういったデータセグメントに分かれているのか？といったことを少しでも理解する手助けにはなりそうですね。

Chapter 8

レコメンデーションの仕組みと実装

Section

01 おすすめ商品をレコメンドして購入回数を向上させよう

ショッピングサイトとか動画配信サービスって、おすすめコンテンツが表示されるじゃないですか。あれってどういう仕組みなんですか？

同じような商品を買った人の傾向を分析して、「その人たちはほかにもこういう商品を買ってる。だからあなたも気に入るはず」っていう商品をおすすめしてくれるレコメンデーションっていう機能よ。

なるほど。そういわれてみるとシンプルですね。

大量の情報があふれる世の中にあって、個人個人に必要な情報をおすすめしてくれるレコメンデーションは、いま研究が盛んな分野の１つなのよ。そのサービスの使いやすさにも直結するしね。

まあ、ときどき余計なお世話に感じることもありますけど……。

それではレコメンデーションの世界を覗いてみましょう！

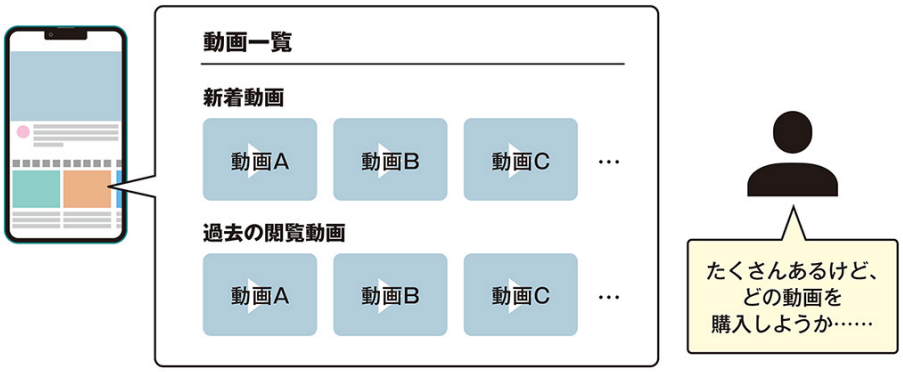
ここで学ぶこと

- レコメンデーションの基本的な仕組み
- ユーザーの類似度を計算する
- コンテンツの類似度を計算する

とあるオンライン動画配信サービスの課題を考えてみよう

とあるオンライン動画配信サービスの運営会社Ｆ社のケースを考えてみましょう※1。近年のオンライン動画配信サービスとしては、Amazon Prime、Netflix、Hulu、TSUTAYA TV といったさまざまサービスが存在します。どのようなサービスであっても、基本的なサービス概要としては、ユーザーに対してさまざまな動画を提供し、ユーザーは自分が見たいと思った動画を閲覧するという流れとなります。

➲ さまざまな動画の中から、どの動画を選択すればよいのか？［図 8-1-1］



Ｆ社のサービスでは、ドラマや映画など多くの動画を取り扱っていることが魅力で、ユーザー数を伸ばしてきました。しかし一方で、動画数が増えてきていることで、**ユーザーは、たくさんある動画の中からどれを選べばよいのかを選択しにくくなっているという課題**が生じてきました。その課題を解決するための施策を考えることにしました。

まず、ビジネスモデルとしては、大きくは以下の２つが存在します。

➲ 動画配信サービスのビジネスモデル［図 8-1-2］

・１動画ごとに従量課金するパターン
・定額料金支払いで、好きなだけ動画を閲覧できるパターン

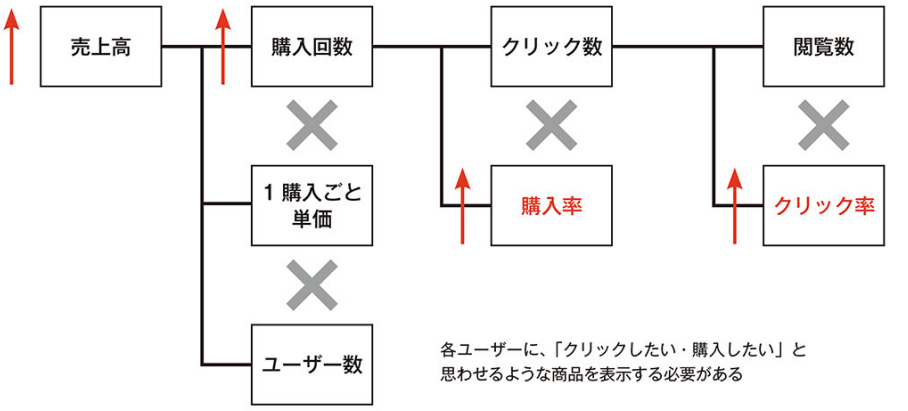
（※1）わかりやすい事例として、Amazon Prime や Netflix のような動画配信サービスを取り上げていますが、一般的な EC サイトなどさまざまなビジネスにおいても、今回学習するレコメンデーションを活用できるでしょう。

どちらのモデルで考えても本質的には同じですが、今回はKPIをシンプルに考えやすいように前者のパターンを想定してみます。どのようなビジネスであれ、売上高を上げることは重要となります。とはいえ、それだけだと施策につながらないので、売上高をさらに細かいKPIに分解します。［図8-1-3］のようにKPIをツリーの枝葉のように段階的に設定したものを**「KPIツリー」**と呼びます。

ユーザーが、たくさんある動画から自分が見たい動画を適切に選ぶことができれば、動画のクリック率や購入率が上がるはずです。すると（ユーザーあたりの）購入回数があがり、結果的に売上高が上がる、という構図です。

➲ 売上高をKPIツリーで表現する［図 8-1-3］

KPIツリー



ユーザーのクリック率や購入率を上げるための施策はたくさんあります。今回はユーザーに適切に動画を選んでもらえるように、**ユーザーごとに、より興味を持つであろう商品をレコメンド（推薦）して、迷いなくコンテンツを選択してもらい、クリック率や購入率を改善**することとしました。

［図8-1-4］はAmazonの例です。まさに過去のデータにもとづいて、ユーザーや商品ごとに「オススメ商品」や「関連商品」が表示されています。よりクリックされやすく、より購入されやすいような状態になっていることがわかります。

このようなユーザーや商品ごとにパーソナライズして、より選んでもらえるようなコンテンツを推薦する施策を**「レコメンデーション」**といい、ビジネス適用が非常に盛んなデータサイエンスの活用方法となっています。

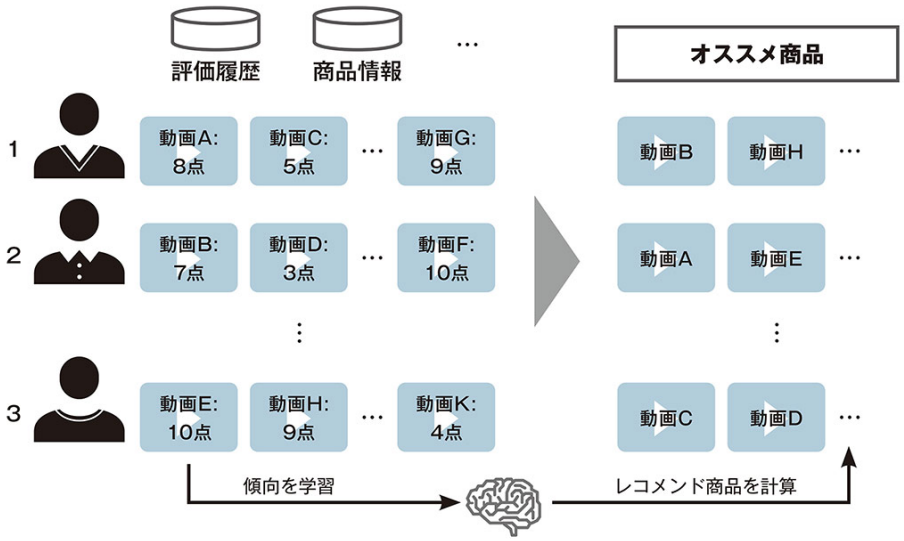
➲（例：Amazon）おすすめ・関連商品を表示する［図 8-1-4］

データサイエンスで解くための問題設定

それではレコメンデーションエンジンを実装していくための設定をより詳細化します。前提としてですが、レコメンデーションの分野はビジネスに適用しやすいということもあり、先端企業を中心に非常に研究が盛んです。したがって「どういったデータやアルゴリズムを用いて、どのようなレコメンドをするか」の方法論は非常に多岐に渡ります。技術的・理論的に難しい内容も多いため、今回は基礎的な内容を紹介します。

たとえばユーザーごとに各動画に対する評価値の履歴データがあったとしましょう（「星いくつ」といったようなレーティングデータをイメージするとよいでしょう）。そのデータから、**どのユーザーがどういった動画に高評価をつけるかという傾向を学習**します。それにもとづき、ユーザーごとに「まだ見ていないが好みであろうおすすめ動画」をレコメンドする、というロジックを構築していきましょう。

➲ ユーザーごとの過去の評価データから傾向を学習し、レコメンド商品を計算［図 8-1-5］



「どういったコンテンツを評価しているか」というデータの定義は、さまざまなパターンが考えられるわね。また、後述するけど、仮にレーティング情報があまりなかったり評価機能がなかったりする場合は、たとえば「そのコンテンツを購入したかどうか」「クリックしたかどうか」といった別の変数を考えて定義する必要もあるということを知っておきましょう。

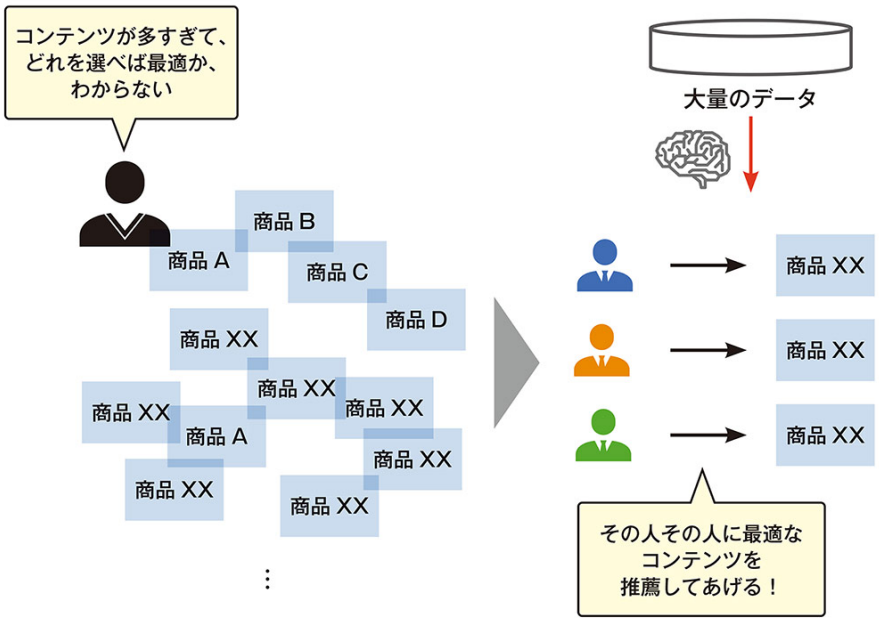
Section

02 レコメンデーションエンジンの概要

なぜレコメンデーションが必要なのか？

本 Section からはレコメンデーションエンジンの概要を説明していきますが、そもそもなぜレコメンデーションが必要なのでしょうか？ 皆さんご存知のように、情報化技術の進展により大量の情報が蓄積・発信されるようになりました。その結果、**情報過多な時代**となり、ユーザーにとって有用な情報を見つけ出すレコメンデーション（推薦）システムが考案されたのです。

➲ **情報過多の時代において、ユーザーに最適な情報を推薦する重要性がある**
[図 8-2-1]



［神嶌 2016］※2 をそのまま転用すると、レコメンデーションシステムは「**どれに価値があるかを特定するのを助ける道具**」と定義されています。つまり、その人その人に最適であろう価値あるコンテンツを推薦することがレコメン

（※ 2）神嶌敏弘『推薦システムのアルゴリズム』（https://www.kamishima.net/archive/recsysdoc.pdf）のこと。

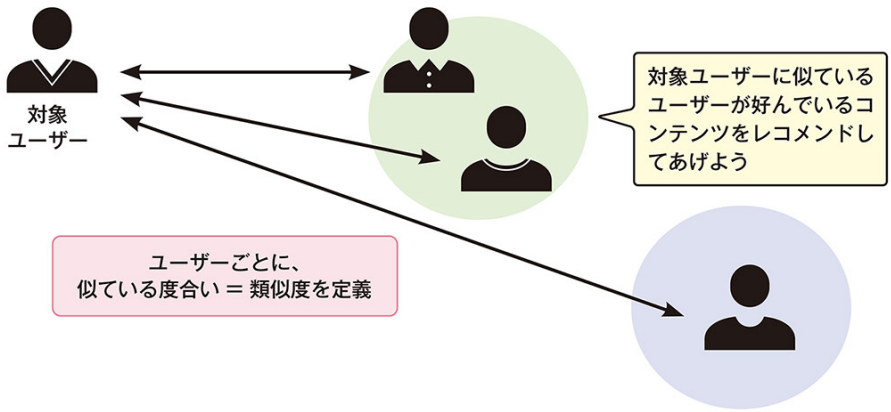
デーションシステムの役目である、と考えることができます。

レコメンデーションの基本的な考え方

レコメンデーションエンジンの必要性が理解できたところで、その考え方を学んでいきましょう。まずはそのコンセプトを改めて押さえておきます。

前述したようにさまざまなロジックが存在しますが、基本的な考え方としては、**あるユーザーに似ているユーザーを算出することで、その似ているユーザーが好んでいるコンテンツがレコメンドすべきものである**、というものです。そして「似ている」という度合いを定量化する必要があるため、何かしらのロジックで**「類似度」**を定義し、過去のデータから類似度を計算する必要があります。そのような考え方にもとづいた基本的なアルゴリズムを「協調フィルタリング」といいます（後述）。

➲ 自分に似ているユーザーの好むコンテンツをレコメンドする［図 8-2-2］

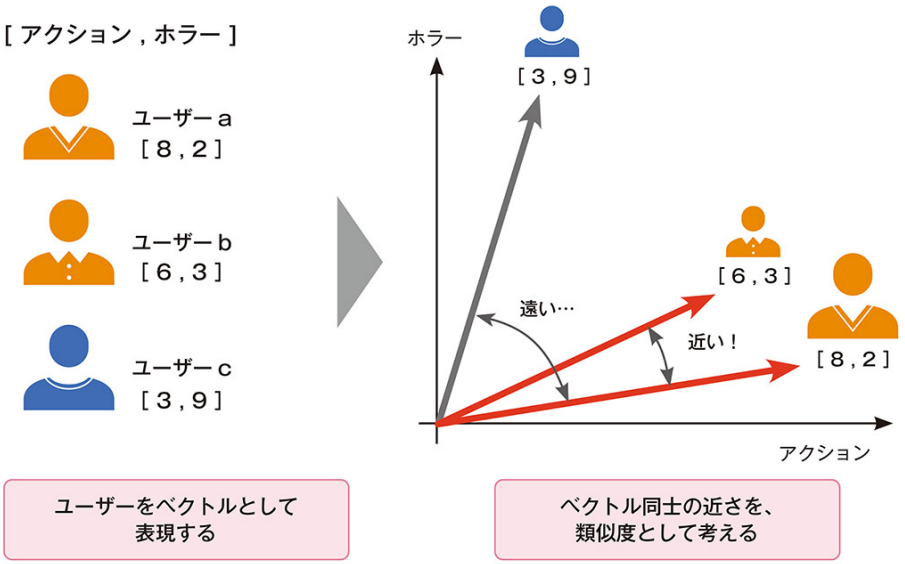


また、ユーザー軸ではなく、コンテンツ同士の類似度を計算することで、ある商品の関連商品（おすすめ商品）をレコメンドする、といった考え方もあります。これは「コンテンツマッチング」と呼ばれますが、本質的にはどちらも**類似度によって似ているものは何かをデータから定義・計算する**、という行為になります。

ベクトル表現により類似度を定義する

前述した「類似度」を定義するのに重要なのが数学の「**ベクトル**」という考え方です。たとえば、アクション系、ホラー系の動画の評価値があるとしましょう。あるユーザー A の評価値がそれぞれアクション系:8、ホラー系:2 だとすると、そのユーザーの評価値のベクトルは［8, 2］という 2 次元のベクトルで表せます。同様にほかのユーザーの評価値ベクトルが［6, 3］や［3, 9］だとします。**ベクトルは「大きさ」と「向き」で表す**ことができるので、それぞれのベクトルは［図 8-2-3］の右図のように表現できます。そして、ベクトル同士の近さを**角度**で測れます。仮にあなたがユーザー a だとすると、あなたは［8, 2］というベクトルを持っているといえます。このとき［8, 2］に対する角度を考えると、［6, 3］のベクトルをもつユーザーは近く、［3, 9］のベクトルをもつユーザーは遠いユーザーだと考えられます。

➲ ユーザーのデータをベクトル化し、ベクトル同士の類似度を考える［図 8-2-3］



このように、ベクトルにより計算できる角度によって、ユーザー同士の類似性を考えることができます。

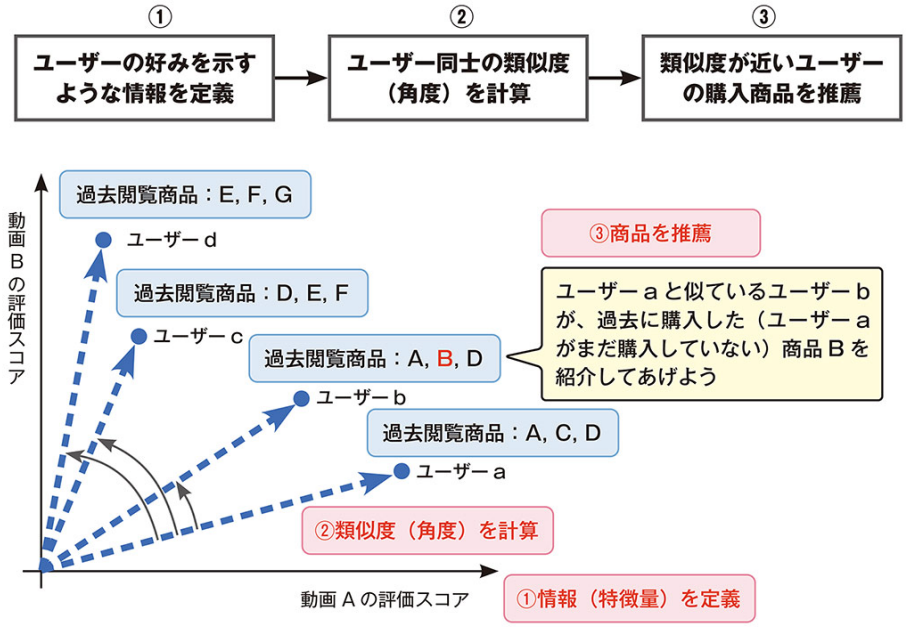
Section

03 ユーザーの嗜好を考慮する「協調フィルタリング」

ユーザーの行動履歴をベクトル化する

それでは、より具体的にレコメンデーションエンジンのロジックを深堀りしていきましょう。本書では、基本的かつ重要な「**協調フィルタリング**」を紹介します。協調フィルタリングは、**ユーザーの好みの傾向を考慮したレコメンド**になります。まず大まかなロジックに関しては、[図 8-3-1] の 3 ステップにまとめることができます。

➲ 協調フィルタリングによるレコメンドの概要［図 8-3-1］



最初の「ユーザーの好みを示すような情報の定義」に関してですが、そのような情報＝特徴量は非常に多岐に渡ります。その中でもわかりやすいもの

の１つが、コンテンツに対する評価値です。ユーザーごとに、動画に対する評価値が［図 8-3-2］のようにあったとしましょう。その評価値ベクトルが、各ユーザーが持っているベクトルであると考えられます。

➲ ユーザーのコンテンツ（動画）に対する評価値をベクトル化する［図 8-3-2］

類似度を「コサイン類似度」で定義する

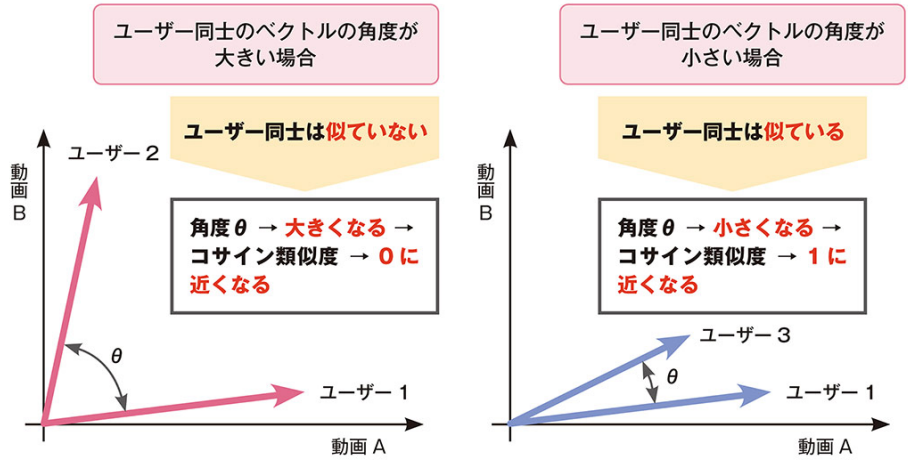
評価値ベクトルをもとに、ユーザー同士の類似度を考えます。実際には、動画のコンテンツ数は非常に多いはずですが、説明しやすいように動画数が２だとしましょう（本質的には３つ以上でも変わりません）。その場合、仮にユーザー同士のベクトルが離れているならば、ベクトル同士の角度θ（シータと読みます）は［図 8-3-3］左図のように大きくなるはずです。

その場合、その近さ度合いを表す有名な指標として「**コサイン類似度**」というものがあります。コサイン類似度は、［図 8-3-3］の左図のように角度θが大きくなると、０に近くなるように定義されています※3。

一方で、もしユーザー同士の評価スコアのベクトルの向きが近ければ、ベクトル同士の角度は小さくなります。するとコサイン類似度は１に近くなります。したがって、**ユーザー同士のベクトルの角度が近くなる＝ユーザー同士の嗜好が似ている場合、コサイン類似度は１に近づくように定義**されています。

（※3）コサインというのは、数学におけるサイン、コサイン、タンジェント（sin、cos、tan）に由来します。コサイン類似度の計算の定義は今回は省略しますが、ベクトル同士の角度をもとに計算されていると考えれば大丈夫です。

➲ 評価値ベクトルをもとに、コサイン類似度を計算できる［図 8-3-3］



ユーザーごとの類似度行列が作成できる

ユーザー同士の似ている度合い＝コサイン類似度が定義できたら、それをすべてのユーザー同士に対して、ユーザー間のコサイン類似度を計算します。すると、ユーザー同士の**類似度を表した行列（類似度行列）**を作ることができます。

そのユーザー数×ユーザー数の類似度行列を作成することで、どのユーザーとどのユーザーが類似しているかが、わかることになります。

➲ ユーザー同士の類似度行列を定義できる［図 8-3-4］

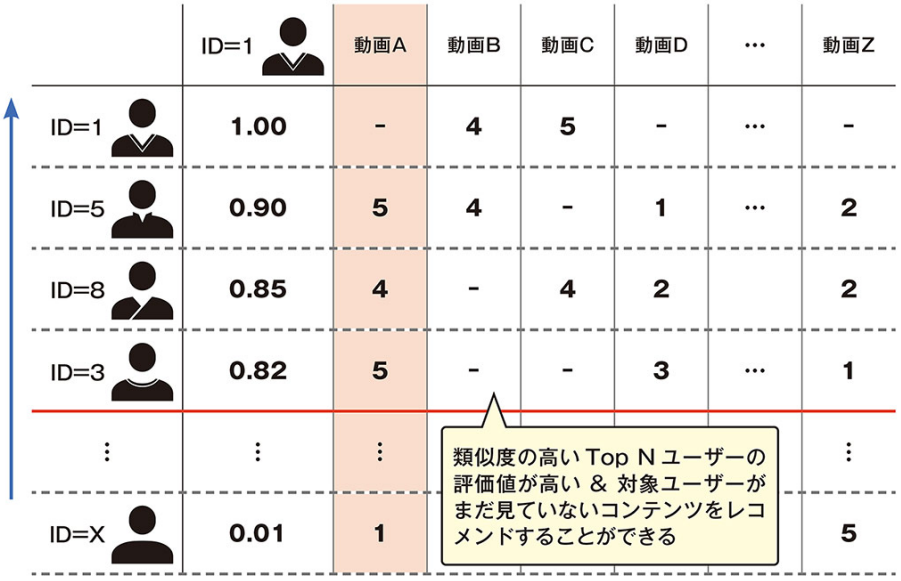
類似度にもとづいてレコメンドコンテンツを計算する

ユーザー同士の類似度スコアを算出できたら、そのスコアをもとに、どのようにレコメンドするかを考えます。前提としては、ロジカルな考え方であればどのような考え方でも正当性があります。

今回は、簡単に考えられるロジックとして、［図 8-3-5］のように考えてみます。たくさんいるユーザーのうち、**類似度の高いトップ N 人を選択し、そのユーザー群の高評価コンテンツ**を、レコメンドすべきコンテンツとすることができるでしょう。

かつ、すでに対象ユーザーが見ているコンテンツをレコメンドする必要はあまりないので、**そのユーザーが見ていない、かつ類似度の高いユーザーが高評価としたコンテンツをレコメンド**する、と考えられます。

➲ 類似度スコアに基づいて、レコメンドすべきコンテンツを考えられる［図 8-3-5］



なお、コンテンツへの評価値が十分に蓄積されていないようなケースもあるでしょう。ユーザーにレーティングしてもらう機能がないような場合です。その場合は、たとえば評価値に変わって、

・購入したかどうか（1 か 0）
・クリックしたかどうか（1 か 0）

といった変数を代用するといったロジックも考えられます。

今回紹介した協調フィルタリングにはいくつか課題もあります。一例として、サービスに登録したばかりのユーザーなどは、評価値などの履歴データが蓄積されていないため、適切にレコメンドができません。そのような現象を「**コールドスタート問題**」と呼びます。

コールドスタート問題への対応策の 1 つとしては、次 Section で紹介するコンテンツマッチングによる方法を駆使する方法が考えられます。

協調フィルタリングは、近年のレコメンデーションエンジンの中では比較的シンプルな手法だけど、現在でも Amazon は協調フィルタリングベースのアルゴリズムを採用しているといわれていて、今なお強力な手法の 1 つといえるわね。もちろん、Amazon のようなビッグサービスになると、このままの適用は難しいので、処理速度を高速化したり、アルゴリズムを細かくチューニングしたりといった改良も必要ね。

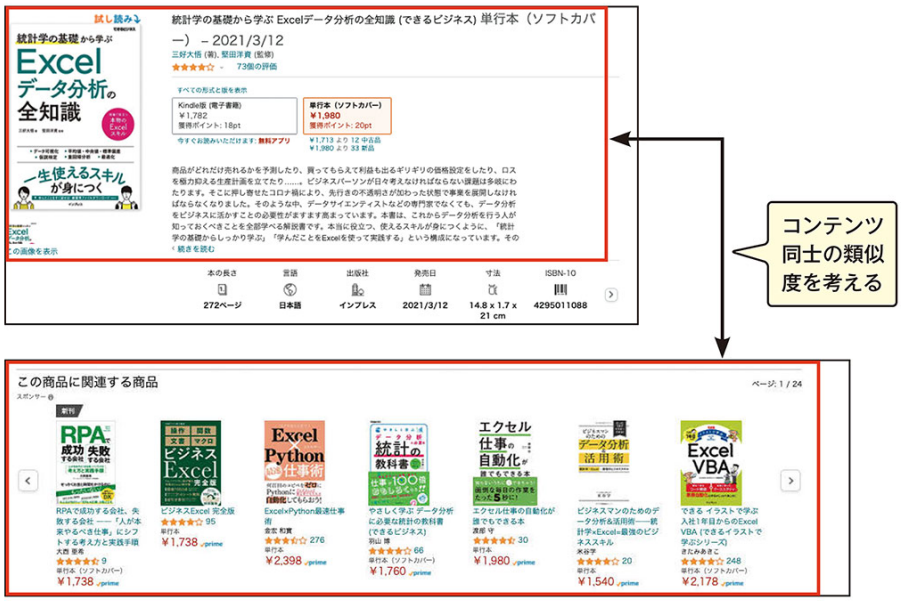
Section

04 コンテンツの内容を考慮する「コンテンツマッチング」

コンテンツの情報をベクトル化する

前 Section では、ユーザーの過去データを用いたレコメンドロジックを考えましたが、ほかにもコンテンツ同士の類似度を直接考えるやり方があります。この方法を「コンテンツマッチング」と呼びます。

➲ コンテンツ同士の類似度を考える［図 8-4-1］



コンテンツマッチングの考え方も、基本的には協調フィルタリングと同じで、コンテンツの情報＝特徴量を定義し、コンテンツのベクトルを得ます。コンテンツの情報はさまざま考えられますが、たとえば映画コンテンツであれば、一例として以下のような情報が考えられるでしょう。

➲ コンテンツ情報の例［図 8-4-2］

・公開年
・興行収入
・ジャンル（アニメ系か、アクション系かなど）
・ユーザーからの平均評価値や、評価された回数
・etc

それらの情報をもとに、［図 8-4-3］のようにコンテンツごとのベクトルを生成できます。

➲ コンテンツの情報からベクトルを生成する［図 8-4-3］

ベクトルを生成することができれば、あとは前 Section と同様に次のようなレコメンドを行えます。

・コンテンツ同士の類似度を（コサイン類似度などで）計算し、
・あるコンテンツに似たコンテンツ群を表示する

このレコメンデーションロジックは新規登録したばかりのユーザーに対してもすぐに適用できるので、上述したコールドスタート問題に対処できます。

また、ある商品詳細ページにおいて、「関連商品」をレコメンドするといったこともできます。

Tips　教師あり学習を活用したレコメンデーション

レコメンデーションエンジンは非常に奥が深く、さまざまな方法論が研究されています。その一例として、前章で学んだ「教師あり学習」を活用したレコメンデーションが近年ではよく利用されます。そのイメージとしては、［図 8-4-4］のようにユーザー×アイテム（コンテンツ）となるように行を定義します。そして、以下のような特徴量を追加したうえで目的変数を定義します。

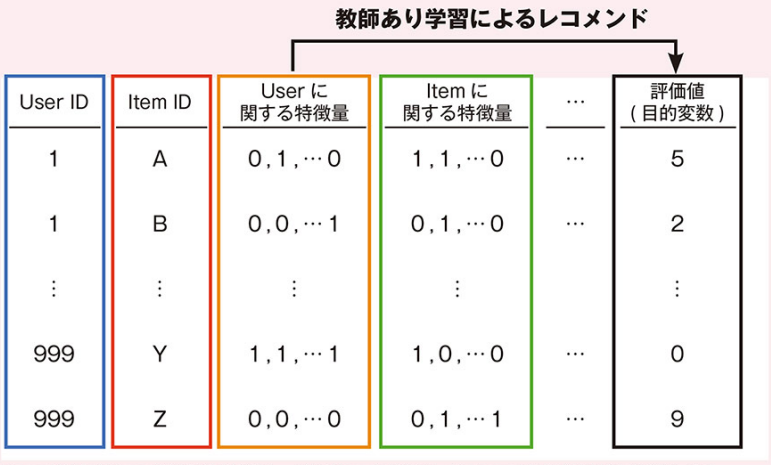
・ユーザーに関する特徴量（どのようなジャンルのコンテンツを何回閲覧したか、など）
・アイテムに関する特徴量（どのようなジャンルのコンテンツか、など）

目的変数としては、たとえば各ユーザーが各アイテムにつけた評価値などが考えられます。

すると、特徴量から目的変数を学習する教師あり学習モデルを構築できます。モデルができれば、あるユーザーにあるアイテムをレコメンドした場合、どのくらいの評価値になりそうかを予測することができます。その際に高評価となりそうなアイテムをレコメンドすればよい、と考えられます。

実際には、考慮しなければならない細かい点がたくさんありますが、このように教師あり学習を用いた考え方もある、ということを理解しておけばよいかと思います。

➲ 教師あり学習によるレコメンデーションエンジンの構築イメージ［図 8-4-4］



User がどんな Item を好むかを学習し、どの User にどういった Item が適切かを予測（推薦）する

Section

05 実践：ユーザー評価データを活用しよう

練習用ファイル：chap08_book_recommendation / items.csv、ratings.csv

実践 データの確認

最後に、実践演習を通してレコメンデーションエンジンのビジネス適用のイメージを深めていきましょう。冒頭で述べたように、今回の検討施策はオンライン動画配信サイトにおいて、**たくさんある動画の中からどれを選べばよいのかを選択しにくくなっているという課題を解決**するために、**ユーザーごとにより興味を持つであろう商品をレコメンド（推薦）して、迷いなくコンテンツを選択してもらい、クリック率や購入率を改善**することです。そこで、まずはシンプルなレコメンデーションエンジンを構築することとします。

今回使用したデータは「chap08_book_recommendation」フォルダ内に格納してあります※4。今回は以下2つのCSVデータを使用します。

・items.csv：商品情報
・ratings.csv：どのユーザーがどの商品にいくつの評価値をつけたか

今回は動画コンテンツを取り扱うため、商品情報も「タイタニック」や「スター・ウォーズ」といった映画コンテンツとなっています。本来は大量の商品データを取り扱いますが、今回は30商品を取り扱い対象としておきましょう（本質的な考え方としては、30商品でも数万商品でも変わりません）。

前Sectionまでに紹介したように、レコメンデーションエンジンといってもさまざまなアルゴリズムが存在します。今回は先ほど学んだ「協調フィルタリング」によるロジックを構築していきましょう。

（※4）今回使用したデータは、MovieLensという有名なオープンデータセットから一部データ抽出をしたものとなります。
https://grouplens.org/datasets/movielens/

➲ レコメンデーションエンジン構築に使用するデータセット［図 8-5-1］

item_id	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
150	Apollo 13 (1995)	Adventure|Drama|IMAX
165	Die Hard: With a Vengeance (1995)	Action|Crime|Thriller
260	Star Wars: Episode IV - A New Hope (1977)	Action|Adventure|Sci-Fi
318	Shawshank Redemption, The (1994)	Crime|Drama
356	Forrest Gump (1994)	Comedy|Drama|Romance|War
364	Lion King, The (1994)	Adventure|Animation|Children|Drama|Musical|IMAX
480	Jurassic Park (1993)	Action|Adventure|Sci-Fi|Thriller
589	Terminator 2: Judgment Day (1991)	Action|Sci-Fi
592	Batman (1989)	Action|Crime|Thriller
595	Beauty and the Beast (1991)	Animation|Children|Fantasy|Musical|Romance|IMAX
648	Mission: Impossible (1996)	Action|Adventure|Mystery|Thriller
780	Independence Day (a.k.a. ID4) (1996)	Action|Adventure|Sci-Fi|Thriller
1036	Die Hard (1988)	Action|Crime|Thriller
1196	Star Wars: Episode V - The Empire Strikes Back (1980)	Action|Adventure|Sci-Fi
1210	Star Wars: Episode VI - Return of the Jedi (1983)	Action|Adventure|Sci-Fi
1240	Terminator, The (1984)	Action|Sci-Fi|Thriller
1270	Back to the Future (1985)	Adventure|Comedy|Sci-Fi
1580	Men in Black (a.k.a. MIB) (1997)	Action|Comedy|Sci-Fi
1721	Titanic (1997)	Drama|Romance
2571	Matrix, The (1999)	Action|Sci-Fi|Thriller
2628	Star Wars: Episode I - The Phantom Menace (1999)	Action|Adventure|Sci-Fi
3793	X-Men (2000)	Action|Adventure|Sci-Fi
4306	Shrek (2001)	Adventure|Animation|Children|Comedy|Fantasy|Romance
4993	Lord of the Rings: The Fellowship of the Ring, The (2001)	Adventure|Fantasy
5349	Spider-Man (2002)	Action|Adventure|Sci-Fi|Thriller

items.csv

- item_id: 商品ID
- title: 商品タイトル
- genres: 商品ジャンル

user_id	item_id	rating
1	1	8
1	260	10
1	356	8
1	480	8
1	592	8
1	648	6
1	780	6
1	1196	10
1	1210	10
1	1240	10
1	1270	10
1	1580	6

ratings.csv

- user_id: ユーザー ID
- item_id: 商品ID
- rating: 評価値 (1~10)

練習用ファイル：chap08_book_recommendation / ratings.csv

実践 ユーザーごとの評価値ベクトルを作成する

今回は、ratings.csv にある評価値（rating）が重要です。この例のユニークユーザー数は 566 となっているので、566 名それぞれに関して、商品ごとの評価値をベクトル化します。また、商品数は 30 なので、ユーザーごとに 30 次元の評価値ベクトルが生成されることとなります。

なお、すべてのユーザーがすべての商品を評価しているわけではないので（それでは何もレコメンドできないですね）、未評価の商品が存在します。**そのような未評価商品の評価値は 0 で補完する形**とします。

今回はわかりやすいようにシンプルに評価値だけを利用する形にしたけど、本来はオンラインサービスなのでクリック履歴などさまざまなデータが存在するはず。そして精度の高いレコメンデーションエンジンを構築しようと思うと、そういったさまざまなデータを用いることになるわね。

➲ ユーザーごとに、各アイテムの評価値ベクトルを生成する［図8-5-2］

練習用ファイル：chap08_book_recommendation / result_similarity_matrix_example.xlsx

実践 ユーザー同士の類似度スコアを算出する

ユーザーのベクトルを定義できれば、続いてベクトル同士の類似度を計算します。類似度にはいくつもの種類がありますが、今回は先ほど学んだコサイン類似度を計算することとします。

566 名すべてのユーザーの類似度をチェックするのは少し骨が折れるので、今回はサンプルとして、10 名分のユーザー同士のコサイン類似度を計算した結果を result_similarity_matrix_example.xlsx に記載しています。ここでは類似度行列として表しているので、行列の上三角（もしくは下三角）部分だけ見れば大丈夫です（[図 8-5-3]）。自分自身とのベクトルは完全一致するので、当然類似度は 1.0 となります。そのほかのユーザーとは、**類似していれば 1 に近づき、類似していないほど 0 に近づきます**。

➲ 10 名のユーザー同士の類似度を表した類似度行列［図8-5-3］

user_id	1	2	4	5	6	7	8	9	10	11
1	1.00	0.00	0.52	0.18	0.29	0.72	0.24	0.17	0.10	0.36
2	0.00	1.00	0.00	0.14	0.16	0.00	0.22	0.00	0.32	0.13
4	0.52	0.00	1.00	0.18	0.12	0.40	0.00	0.00	0.01	0.10
5	0.18	0.14	0.18	1.00	0.66	0.36	0.55	0.00	0.00	0.31
6	0.29	0.16	0.12	0.66	1.00	0.51	0.79	0.00	0.13	0.68
7	0.72	0.00	0.40	0.36	0.51	1.00	0.32	0.42	0.44	0.54
8	0.24	0.22	0.00	0.55	0.79	0.32	1.00	0.00	0.11	0.53
9	0.17	0.00	0.00	0.00	0.00	0.42	0.00	1.00	0.48	0.00
10	0.10	0.32	0.01	0.00	0.13	0.44	0.11	0.48	1.00	0.14
11	0.36	0.13	0.10	0.31	0.68	0.54	0.53	0.00	0.14	1.00

ユーザーの評価値ベクトルから、ユーザー同士の類似度行列が計算できる

練習用ファイル：chap08_book_recommendation / result_recommend_all.xlsx

実践 レコメンドすべき商品を計算する

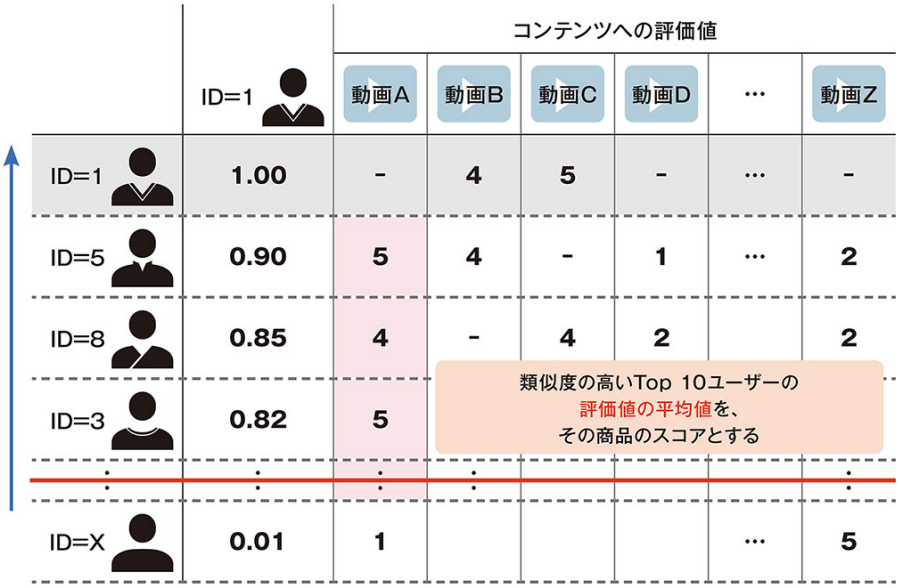
ユーザー同士の類似度が計算できたら、レコメンドすべき商品を計算します。ここでもさまざまな考え方がありますが、今回はわかりやすさを重視して、シンプルに以下のロジックでレコメンド商品を計算してみます。

➲ レコメンドのロジック［図 8-5-4］

- あるユーザーに対して、類似度の高い Top10 ユーザーを抽出
- それぞれの商品に関して、その Top10 ユーザーの評価値の平均値をレコメンドスコアとして計算
- そのスコアが大きい商品の順に、最大 10 商品をレコメンドする
- ただし、そのユーザーがすでに評価している商品は除外する

上記のロジックを平たく表せば、**自分に似ているユーザーが高評価をつけた商品がレコメンドされる**、といえるでしょう。

➲ 類似度の近いトップ 10 ユーザーの評価値の平均値を計算する［図 8-5-5］



そのロジックで計算した結果を result_recommend_all.xlsx に記載しています。それぞれのユーザー（user_id）に対して、上記のロジックで計算されたスコア（score）順に商品が並んでいます。

➲ それぞれのユーザーにレコメンドすべき商品をリストアップ［図8-5-6］

user_id	item_id	score	title	genres
1	318	9.333333333	Shawshank Redemption, The (1994)	Crime|Drama
1	595	9	Beauty and the Beast (1991)	Animation|Children|Fantasy|Musical|Romance|IMAX
1	589	8.5	Terminator 2: Judgment Day (1991)	Action|Sci-Fi
1	4306	8	Shrek (2001)	Adventure|Animation|Children|Comedy|Fantasy|Romance
1	1036	7.875	Die Hard (1988)	Action|Crime|Thriller
1	364	7.5	Lion King, The (1994)	Adve
1	150	7.333333333	Apollo 13 (1995)	Adve
1	8961	7	Incredibles, The (2004)	Actic
1	58559	7	Dark Knight, The (2008)	Actio.
1	79132	7	Inception (2010)	Action|Crime|Drama|Mystery|Sci-Fi|Thriller|IMAX
2	5952	10	Lord of the Rings: The Two Towers, The (2002)	Adventure|Fantasy
2	356	9	Forrest Gump (1994)	Comedy|Drama|Romance|War
2	2571	8.5	Matrix, The (1999)	Action|Sci-Fi|Thriller
2	4993	8.5	Lord of the Rings: The Fellowship of the Ring, The (2001)	Adve
2	1	8	Toy Story (1995)	Adve
2	260	8	Star Wars: Episode IV - A New Hope (1977)	Actic
2	3793	7	X-Men (2000)	Actic
2	1036	6	Die Hard (1988)	Action|Crime|Thriller
4	4993	9	Lord of the Rings: The Fellowship of the Ring, The (2001)	Adventure|Fantasy
4	589	8.5	Terminator 2: Judgment Day (1991)	Action|Sci-Fi
4	1	8	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
4	318	8	Shawshank Redemption, The (1994)	Crime|Drama
4	58559	8	Dark Knight, The (2008)	Action|Crime|Drama|IMAX
4	480	7	Jurassic Park (1993)	Actio
4	780	7	Independence Day (a.k.a. ID4) (1996)	Actic
4	1036	6.666666667	Die Hard (1988)	Actic
4	1210	6.5	Star Wars: Episode VI - Return of the Jedi (1983)	Actic
4	150	6	Apollo 13 (1995)	Adventure|Drama|IMAX
5	648	10	Mission: Impossible (1996)	Action|Adventure|Mystery|Thriller
5	356	9.333333333	Forrest Gump (1994)	Com
5	165	7.5	Die Hard: With a Vengeance (1995)	Actic

user_id = 1 へのレコメンド商品

user_id = 2 へのレコメンド商品

user_id = 4 へのレコメンド商品

user_id = 5 へのレコメンド商品

これで出力自体は完成ですが、これだと少しわかりにくいので、個別のユーザーに対して考察してみましょう。

練習用ファイル：chap08_book_recommendation / result_recommend_examples.xlsx

実践 個別ユーザーに対するレコメンド結果を考察

ここでも例として一部のユーザーに着目してみましょう。result_recommend_examples.xlsx に記載をしています。まずは、user_id が 2、134 のユーザーを見てみましょう。「user_2」「user_134」シートに記載があります。

➲ user_id=2, 134 のユーザーに対するレコメンド結果［図8-5-7］

user_id = 2 におけるレコメンド結果

type	item_id	score	title	genres
既に閲覧したコンテンツ	58559	9	Dark Knight, The (2008)	Action|Crime|Drama|IMAX
既に閲覧したコンテンツ	79132	8	Inception (2010)	Action|Crime|Drama|Mystery|Sci-Fi|Thriller|IMAX
既に閲覧したコンテンツ	318	6	Shawshank Redemption, The (1994)	Crime|Drama
レコメンドコンテンツ	5952	10	Lord of the Rings: The Two Towers, The (2002)	Adventure|Fantasy
レコメンドコンテンツ	356	9	Forrest Gump (1994)	Comedy|Drama|Romance|War
レコメンドコンテンツ	2571	8.5	Matrix, The (1999)	Action|Sci-Fi|Thriller
レコメンドコンテンツ	4993	8.5	Lord of the Rings: The Fellowship of the Ring, The (2001)	Adventure|Fantasy
レコメンドコンテンツ	1	8	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
レコメンドコンテンツ	260	8	Star Wars: Episode IV - A New Hope (1977)	Action|Adventure|Sci-Fi
レコメンドコンテンツ	3793	7	X-Men (2000)	Action|Adventure|Sci-Fi
レコメンドコンテンツ	1036	6	Die Hard (1988)	Action|Crime|Thriller

user_id = 134 におけるレコメンド結果

type	item_id	score	title	genres
既に閲覧したコンテンツ	595	10	Beauty and the Beast (1991)	Animation|Children|Fantasy|Musical|Romance|IMAX
既に閲覧したコンテンツ	1	6	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
既に閲覧したコンテンツ	364	6	Lion King, The (1994)	Adventure|Animation|Children|Drama|Musical|IMAX
既に閲覧したコンテンツ	592	6	Batman (1989)	Action|Crime|Thriller
レコメンドコンテンツ	480	8	Jurassic Park (1993)	Action|Adventure|Sci-Fi|Thriller
レコメンドコンテンツ	589	8	Terminator 2: Judgment Day (1991)	Action|Sci-Fi
レコメンドコンテンツ	3793	8	X-Men (2000)	Action|Adventure|Sci-Fi
レコメンドコンテンツ	5952	8	Lord of the Rings: The Two Towers, The (2002)	Adventure|Fantasy
レコメンドコンテンツ	165	7.667	Die Hard: With a Vengeance (1995)	Action|Crime|Thriller
レコメンドコンテンツ	318	7.667	Shawshank Redemption, The (1994)	Crime|Drama
レコメンドコンテンツ	150	7.6	Apollo 13 (1995)	Adventure|Drama|IMAX
レコメンドコンテンツ	1721	6	Titanic (1997)	Drama|Romance
レコメンドコンテンツ	648	4	Mission: Impossible (1996)	Action|Adventure|Mystery|Thriller

過去に閲覧したコンテンツ以外で、ユーザー評価傾向に近いコンテンツをレコメンド

user_id=2 のユーザーは「ダークナイト」や「インセプション」といった少し暗めのアクション系映画に高評価を、また「ショーシャンクの空に」に中程度の評価をつけています。そしてレコメンド計算結果をみると「ロード・オブ・ザ・リング」2 作品、ほかに「フォレスト・ガンプ」などの映画もレコメンドされていることがわかります。

また user_id=134 のユーザーは、「美女と野獣」「トイ・ストーリー」「ラ

イオンキング」といったディズニーシリーズ、アドベンチャーものに評価をつけています。そしてレコメンド計算結果をみると「ジュラシック・パーク」「ターミネーター」「X-Men」といったアクションアドベンチャーなどの映画がレコメンドされていることがわかります。

これらのユーザーに関しては、（私の感覚ももちろん入っていますが……）比較的レコメンドの結果に妥当性や納得感がありそうな気がしますね！

user_id が 13、151 のユーザーのレコメンド結果も見てみましょう。「user_13」「user_151」シートに記載があります。

➲ user_id=13, 151 のユーザーに対するレコメンド結果［図8-5-8］

user_id ＝ 13 におけるレコメンド結果

type	item_id	score	title	genres
既に閲覧したコンテンツ	2571	10	Matrix, The (1999)	Action|Sci-Fi|Thriller
既に閲覧したコンテンツ	1721	8	Titanic (1997)	Drama|Romance
既に閲覧したコンテンツ	3793	8	X-Men (2000)	Action|Adventure|Sci-Fi
レコメンドコンテンツ	260	10	Star Wars: Episode IV - A New Hope (1977)	Action|Adventure|Sci-Fi
レコメンドコンテンツ	1196	10	Star Wars: Episode V - The Empire Strikes Back (1980)	Action|Adventure|Sci-Fi
レコメンドコンテンツ	1210	10	Star Wars: Episode VI - Return of the Jedi (1983)	Action|Adventure|Sci-Fi
レコメンドコンテンツ	1580	10	Men in Black (a.k.a. MIB) (1997)	Action|Comedy|Sci-Fi
レコメンドコンテンツ	4306	10	Shrek (2001)	Adventure|Animation|Children|Comedy|Fantasy|Romance
レコメンドコンテンツ	4993	9.5	Lord of the Rings: The Fellowship of the Ring, The (2001)	Adventure|Fantasy
レコメンドコンテンツ	589	8	Terminator 2: Judgment Day (1991)	Action|Sci-Fi
レコメンドコンテンツ	592	8	Batman (1989)	Action|Crime|Thriller
レコメンドコンテンツ	1240	8	Terminator, The (1984)	Action|Sci-Fi|Thriller
レコメンドコンテンツ	5349	8	Spider-Man (2002)	Action|Adventure|Sci-Fi|Thriller

同じシリーズ作品ばかりを
レコメンドしてしまう？…

user_id ＝ 151 におけるレコメンド結果

type	item_id	score	title	genres
既に閲覧したコンテンツ	1	10	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
既に閲覧したコンテンツ	648	6	Mission: Impossible (1996)	Action|Adventure|Mystery|Thriller
既に閲覧したコンテンツ	780	6	Independence Day (a.k.a. ID4) (1996)	Action|Adventure|Sci-Fi|Thriller
レコメンドコンテンツ	260	9.33	Star Wars: Episode IV - A New Hope (1977)	Action|Adventure|Sci-Fi
レコメンドコンテンツ	1210	6	Star Wars: Episode VI - Return of the Jedi (1983)	Action|Adventure|Sci-Fi

レコメンド候補がそもそも
少ない？…

user_id=13 のユーザーは、「マトリックス」「X-Men」といった、SF アクションものに評価をつけています。そしてレコメンド計算結果をみると、同じく SF アドベンチャー系の「スター・ウォーズ」シリーズがレコメンドされているが、同じシリーズがトップにレコメンドされすぎており、少し**レコメンドが偏っている印象を受けてしまう**かもしれません（よりバラエティーのある商品群がレコメンドされるとうれしいですよね）。

また user_id=151 のユーザーは、「トイ・ストーリー」に高評価、次いで「ミッション：インポッシブル」「インデペンデンス・デイ」といったアクション

アドベンチャーものに評価をつけています。そしてレコメンド計算結果をみると、同じ系統の「スター・ウォーズ」シリーズが推薦されているが、2件しかレコメンドされていないことがわかります。これは、user_id=151 に**類似したユーザーが、みな同じような映画ばかり評価していた**ため、レコメンドできる商品が少なかったことが原因と考えられます。実はこのように、協調フィルタリングは、類似しているユーザーの傾向をつかんだレコメンド結果となりますが、一方で**「まだ興味がありそうかはわからないが、試しに類似ユーザーも見ていないような商品を提示してみよう」といったレコメンドは難しい**[※5]点も課題点として挙げられます。

少し応用的な話になりますが、何かを探しているときに、探しているものとは別の価値あるものを見つけることで得られる幸福感を「**セレンディピティ**」と呼びます。セレンディピティは実店舗で買い回っているようなときに起こりますが、今回のような協調フィルタリングだとセレンディピティを演出できないことは1つの課題であると考えられています。近年のレコメンデーションでは、こういった点も考慮しながらアルゴリズムが研究されており、オンラインショッピングにおいても、実世界のショッピング体験を持ち込めるような工夫が重要となってくるでしょう。

レコメンデーションにおける精度評価

これまでの教師あり学習や画像分類の際には、精度評価指標にもとづいて、構築したモデルの精度を評価しました。一方で、レコメンデーションエンジンにおける精度評価は、実は正確に評価できません。なぜならば、評価する時点のデータはすでに購入された／されていないが決まってしまっており、レコメンドをしたかどうかで購入の有無が変わるかを評価できないためです。したがって、（第5章でも紹介しましたが）後述するABテストによる検証で評価したいところです。このような実サービスにおいて実運用に影響がある形で行う検証を「**オンライン検証**」と呼びます。

とはいえ、オンライン検証は実サービスに組み込んで実験する必要があるため、ハードルが高いです。そこで、これまで学んだように、過去データを用いて可能な範囲で精度評価検証をする「**オフライン検証**」の方法論も存在

(※5) もちろん、計算アルゴリズムを色々と工夫することである程度解決を見込むことはできますが、今回紹介したようなシンプルな協調フィルタリングでは難しいでしょう。

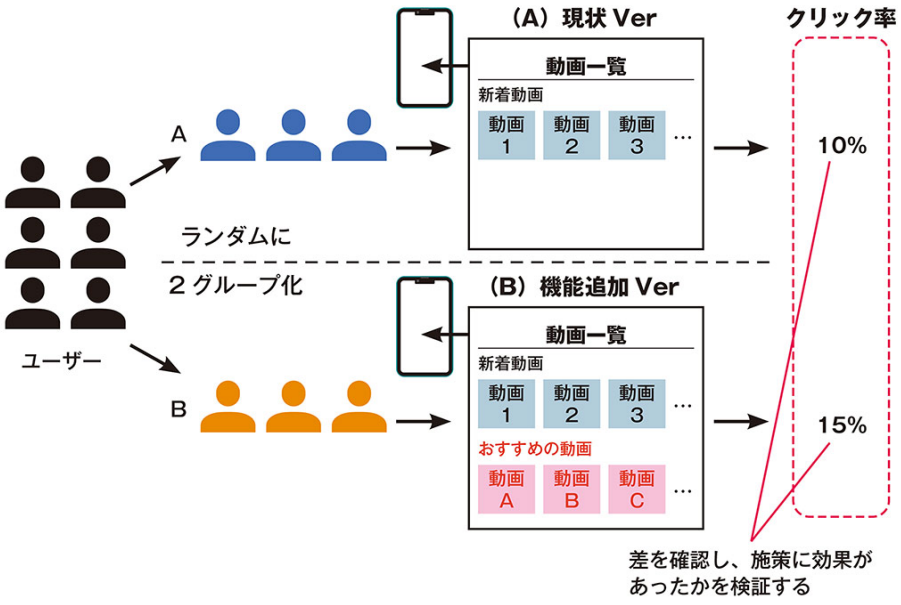
します。レコメンデーションエンジンにおけるオフライン検証時の評価指標は非常に多岐に渡り、どのようにサービス上に実装するかによっても変わってくるので、今回は紹介を省略しますが、オンラインとオフライン両方の検証があるということを意識しておきましょう。

ビジネス上のKPIを効果検証する

オンライン検証ができる場合は、第6章で学んだ**ABテストにより効果検証を行う**のが一般的です。ABテストの考え方はすでに第6章にて紹介しているので、忘れてしまった方は再度見返しておいてください。

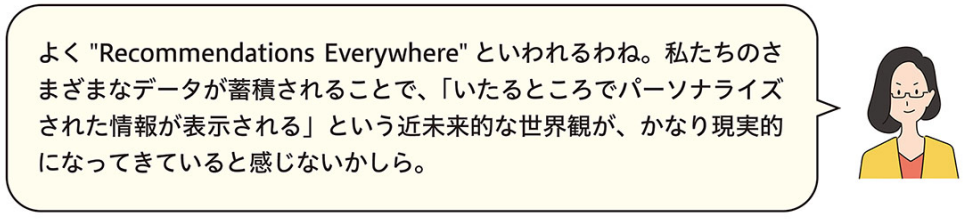
今回の施策は、KPIとして**商品のクリック率や購入率を改善する**ことが1つの目的です。したがって、ランダム分割したユーザーA群とB群に関して、たとえば「A群にはレコメンデーションなし、B群にはレコメンデーションによるおすすめ商品の表示あり」とした場合に、クリック率や購入率がどの程度変わるかを比較します。そして**統計的仮説検定**を用いて、統計的に考えても、きちんと差があるかどうかをチェックすることで、効果があったかどうかを見極めることができます。

➲ ABテストで効果検証をする［図8-5-9］



また、もしすでにレコメンデーションエンジンが導入されているような場合も、たとえばエンジンのアルゴリズムを改良してみたので、どのくらいよくなったかを検証したいでしょう。そのようなときも、同様にABテストとして、旧アルゴリズムと新アルゴリズムで比較することで、本当にアルゴリズムの改良がビジネスKPIに影響を与えたのかを検証することができます。

冒頭含め、何度か述べましたが、レコメンデーションエンジンは施策としてもわかりやすく、また購入点数やCV数の増加など売上に直結しやすい施策なため、昨今のビジネスシーンでは非常によく活用されています。ユーザーとしても、数ある商品の中から、自動的にオススメの商品を推薦してくれることは恩恵があるでしょう。 そういった背景もあり、最近ではレコメンデーションエンジンを提供するサービス（SaaS、Software as a Service）も出てきています。事業側は購買履歴やユーザー情報などのデータをSaaS側にデータ連携し、SaaS内部のレコメンデーションエンジンがその結果をフロントサービスに連携する、というイメージです。一方で、そういったSaaSのような外部サービスを使わずに内製的にエンジンを開発するパターンもあり、どちらがよいかというのは非常に難しい話です（これはレコメンデーションに限らずそのほかの章で紹介した技術でも同様です）。 内製化できるのが望ましいですが、そういった人材などのリソースが足りないケースも多いので、最初のうちは外部サービスを利用して、効果が出そうだったら内製化に向けて開発を進める、またいきなり人材を採用するのではなく、最初は外部パートナーをうまく利用しつつ、伸びていけば、そういった開発に強い人材を雇用する、といった形で、うまく状況に応じてエンジン開発を進めていくことが重要になるでしょう。



■ ここで学んだ重要トピック

- レコメンデーションエンジン
- ベクトル
- コサイン類似度
- 協調フィルタリング
- コールドスタート問題
- コンテンツマッチング
- AB テスト

■ ステップアップにつながるトピック

- ジャッカード係数、ユークリッド距離 など
- オフライン検証（MAP、nDCG など）
- 教師あり学習を活用したレコメンド

レコメンデーションって、ふだんユーザーとしてなにげなく利用していましたが、いろんな仕組みが動いていたんですね。

そうね。レコメンデーションと一口にいっても、サイトのトップ画面と商品詳細画面で、見せるべき商品が変わってくるはずよね。だからとても奥深い分野なの。ユーザーの嗜好を刺激するにはどうすればいいかという感覚を持つことが重要ね。

Chapter 9

数理最適化で利益の最大化を図る

Section

01 商品単価を最適化して利益を最大化しよう

いま担当している小売店の施策なんですけど、販売数アップを目論んでギリギリの価格で攻めていたんですが、クライアントから「そんな単純な施策ではこのあと先細りするのが目に見えている」って言われちゃいました。

それはそうね。販売数よりも「どれだけ利益を稼げるか」を考えないといけないわね。

利益を上げるってことは、仕入れを安くするか販売価格を高くするかしかないじゃないですか。これ以上は安く仕入れられないし、販売価格を上げたら売れなくなりそうだし、ジレンマに陥ってるんです……。

さっき「ギリギリの安い価格で攻めてる」って言ってたじゃない？ なら逆に、利益が最大化するような最適な価格がどこなのかを見つければ解決できそうね。

たしかにそうですが、「最適な価格」なんてぼくにはわかりませんよ……。

それを解決するのが「最適化」よ。いろんなところで使える概念だからこの機会に学んでおくといいわ！

ここで学ぶこと

- ☑ 最適化の仕組みを考える
- ☑ 連続最適化のビジネス活用事例
- ☑ 組み合わせ最適化のビジネス活用事例

とある小売店の課題を考えてみよう

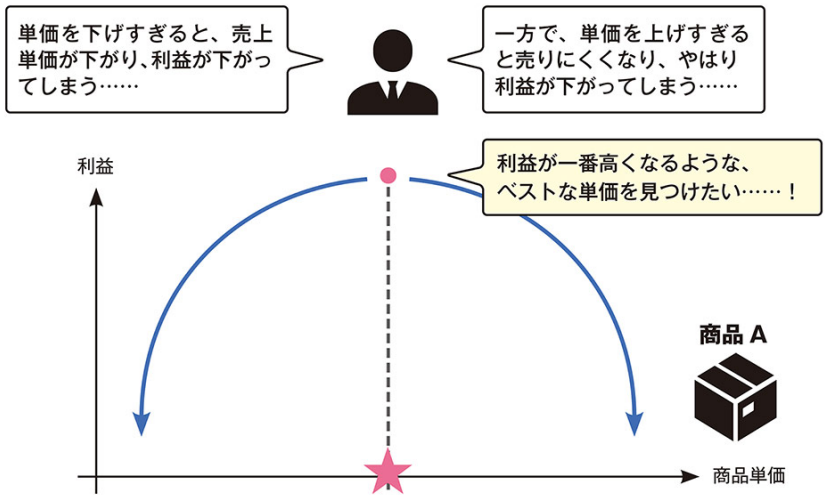
とある小売店の運営会社Ｇ社のケースを考えてみましょう[※1]。Ｇ社運営の小売店では、さまざまな商品を仕入れて販売していますが、取り扱う商品数が増えてきて、１つ１つの商品に対する取り扱いの精度が悪くなってきたことが１つの問題として浮上してきました。

その中で、**商品の価格の適正化**をすべきなのではないかという声が上がってきました。過去の傾向を見ると、（もちろん商品特性によって異なりますが）比較的多くの商品に関して、以下のような現象が見受けられるようになってきています。

・単価を下げすぎると、（当然ですが）売上単価が下がる
・単価を上げすぎると、今度は販売個数が下がる

そこで、**利益が最大となるような商品単価を見つけられないか？**という課題解決を模索することに決まりました。

➲ 利益を最大化するような商品単価を見つけることはできないか？［図9-1-1］



（※1）同様の課題を持つ、たとえば飲食店や、自社商品を持つ事業やサービスなどにおいても、同じような活用が考えられます。

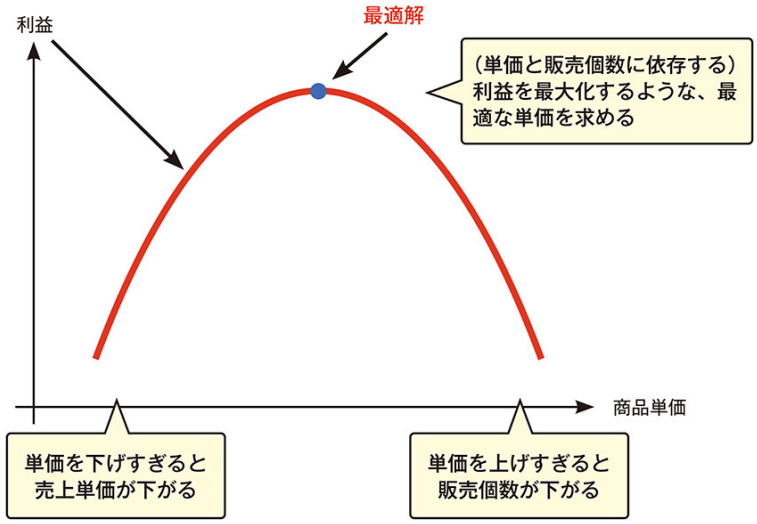
データサイエンスで解くための問題設定

この課題をデータサイエンスで解決するために、「**数理最適化**」という技術分野を利用します。数理最適化の詳細は次 Section 以降で紹介していきますが、これまで紹介してきた教師あり学習などとは、少し毛色が異なる技術となります※2。

今回は、商品単価を変化させて利益を最大化させることがゴールとなります。したがって単価と利益の関係性を、たとえば［図 9-1-2］のように「数式的に」表すことができれば、その数式にもとづいて最適な単価を見つけられそうです。

次 Section からは、まず最適化の概念やビジネスにおける事例を紹介します。そして最後に、この問題を簡単な形で実際に解いて理解を深めていきましょう。

➲ **商品単価に対する利益の関係性を考慮し、最適単価を算出できないか**［図 9-1-2］



（※ 2）本質的にはこれまで紹介した技術と密接に絡み合っている部分も多いのですが、技術的に詳細な内容まで踏み込む必要があるため、本書では詳細は割愛します。

Section

02 最適化の概要

数理最適化とは？

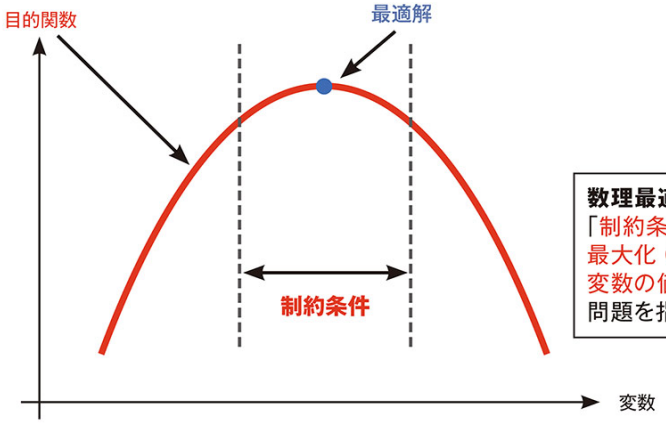
さて、それでは今回の課題を解くための技術を理解していきます。まず、本書で学ぶ「最適化」とは学問的には「**数理最適化**」と呼ばれます（本書では、以降「最適化」という単語も使用していきますが、それは数理最適化のことを指します。）。

数理最適化とは、一言で表すと、「**ある対象となる変数をいろいろと動かしていき、変数によって動く決められた目的関数を最大化（あるいは最小化）するような最適解を求めること**」です。

➲ 数理最適化の定義［図 9-2-1］

数理最適化のフレーム

何を最大化／最小化するか？：「目的関数」と呼ばれる数式を最大化／最小化する
何を変えることで最大化／最小化できるのか？：「変数」と呼ばれるレバーを動かす
最大化／最小化する際の制限は？：「制約条件」と呼ばれる条件式に従う



数理最適化とは、
「制約条件下で目的関数が最大化（or 最小化）する変数の値(＝最適解）を探す」問題を指す

まず最適化の際に絶対に決めなければならないものは、「変数」と「目的関数」です。変数は「最適化によって何の最適解を知りたいのか？」の「何」に当たる部分です。たとえば広告の最適化といっても、「媒体別の広告の出稿金額を知りたいのか？」「ユーザーごとに表示する広告の順番を知りたいのか？」などいろいろある中で、どの値を最適なものにしたいか、具体的に数値で決めておく必要があります。

また変数だけ決めたとしても、何を基準に一番よい値とするかが決まらないと始まりません。それが「目的関数」で、決めた変数の値によって動かされる数式となります。前ページの**［図 9-2-1］では、グラフの横軸が変数、グラフの縦軸が目的関数**となります。そして最適化とは、**「目的関数が最大化（もしくは最小化）するような変数の値はどれか？」**を探し当てるゲームです。そのような変数の値は「**最適解**」と呼ばれます。つまり最適化によって最適解を探し当てるということです。当然、目的「関数」なので、目的関数も変数と同様にすべて数式で表現できなければいけません。

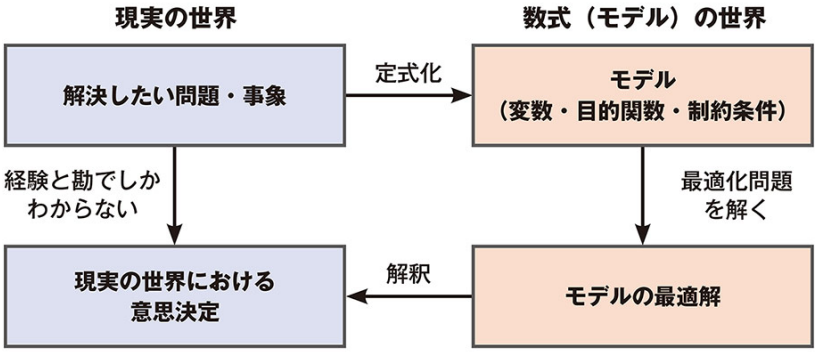
さらに、もう１つ考えないといけない点があります。それは「**制約条件**」です。変数がどんな値でも取れればよいのですが、**多くの場合は「この範囲の中で考えてください」といった制約条件**が課されます。そしてこの制約も数式で表現されている必要があります。 つまり、最適化の定義をこれまでの用語を使って表すと、「**制約条件を満たす範囲で、目的関数を最大化（もしくは最小化）するような変数の最適解を探す**」と言い換えられます。

現実世界の事象を数式（モデル）に落とし込む

これは、「**現実世界の解決したい問題や事象を、何とかして数式のみで表されるモデルの世界に持ち込む。そしてその数式によって定義された最適化問題を解いて得られた最適解を、現実世界の意思決定に応用する**」という営みだともいえます。数式で定義された最適化問題を解くことで、正確な 最適解を導くことができ、最終的にその解で現実世界の意思決定をサポートできるということです。「現実世界で解決したい問題や事象を、いかにして数式だけのモデルの世界に落とし込めるか」ができないと最適化は解けなくなってしまいます。そのモデルの世界に落とし込む際に、最適化の場合は「変

数、目的関数、制約条件」という構造で考えればよいこととなります。

➲ 現実の世界と数式（モデル）の世界 [図 9-2-2]



実際にどのように最適化問題を解くのかは、後半の演習 Section にて、具体的に理解を深めていくこととします。

さて、「最適化」には大きく**「連続最適化」と「組み合わせ最適化」の 2 種類があります**。それぞれの概念を理解しておくことで、ビジネスケースにおいてどう使い分けるのか？というイメージが少しでも深められるはずです。そこで、個別具体の今回の課題に取り組む前に、この 2 つの最適化に関して簡単に紹介しておきましょう。

数理最適化の場合は特に、「要件からどのように最適化式に落とし込むか」という点をしっかりと考える必要があるけど、ここまで学んできた統計学や機械学習も、「現実世界の事象をデータと数式を使って理解する」という点では同じね。つまり、データサイエンスの技術はすべからく、「現実事象を数式（モデル）に落とし込むものである」と考えられるわね。

Section

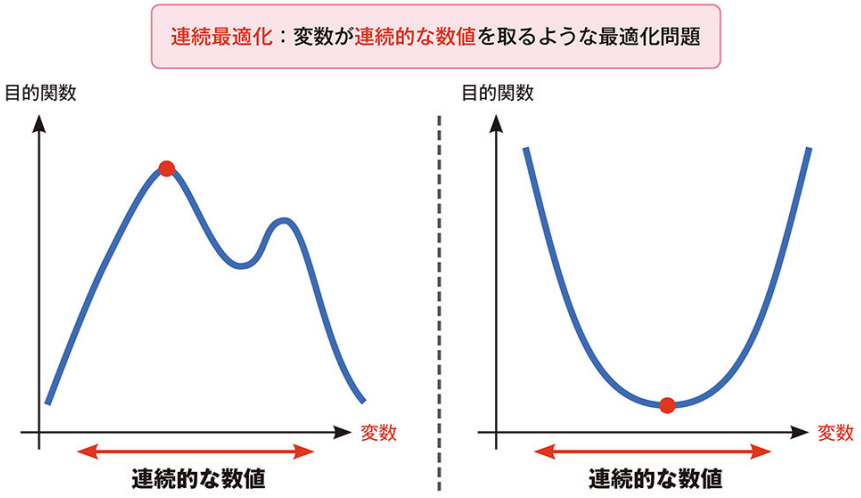
03 2つの最適化①「連続最適化」

連続最適化とは？

連続最適化と組み合わせ最適化の違いは、**変数の種類**にあります。連続最適化とはその名の通り、**変数が連続的な数値を取るような最適化問題**を指します。連続変数とは、0、0.1、0.2、……1.0、1.1……といったつながった値を取る変数です。

イメージとしては、［図 9-3-1］のように、変数は無数の候補を取り、それに応じて目的関数の値が計算されるような形となります。したがって、実は今回の課題で取り上げる商品単価も連続変数に相当します※3。

➲ 連続最適化の定義［図 9-3-1］



（※3）正確にいえば、商品単価自体は小数値は取らない正の整数なので、完全なる連続変数とは言い難いですが、仮に「100.3」という最適解が出れば、四捨五入等で「100」とすればよいので、実務的には連続変数として扱うことが多いです。

ビジネスにおける連続最適化の事例

ビジネスにおいてよく使われる連続最適化の事例を紹介しておきましょう。もちろん、さまざまな活用が考えられますが、たとえば［図 9-3-2］に記載したようなビジネス活用事例があるでしょう。

➲ 連続最適化のビジネス活用例の一例［図 9-3-2］

一例	どのような変数を最適化するか	どのような目的関数を最大 (最小) 化するか
商品価格の最適化	商品ごとの**単価**	**利益**（あるいは売上など）を最大化
広告予算配分の最適化	媒体ごとに対する**出稿金額**	**広告費用 対 効果**（広告経由の売上）を最大化
金融資産ポートフォリオの最適化	銘柄ごとに対する**資産投資金額**	**リスク 対 投資収益**を最大化

今回課題として取り上げた商品価格の最適化は、もう少し発展的な表現をすると「**ダイナミックプライシング**」と呼ばれます。これは文字通りダイナミック（動的に）にプライシング（価格最適化）するというもので、たとえばユニバーサル・スタジオ・ジャパン（USJ）のチケット代は、最近では需要や環境状態に応じて日々刻々と変動していますが、それに近いイメージを持つとわかりやすいでしょう。ただ実際に日々刻々と変動するようなダイナミックプライシングをしようと思うと、システム設計なども複雑に絡んできて難易度がぐっと上がります。そのため、今回の課題例は商品単価の「見直し」程度に捉えておきましょう。

また価格最適化以外にもよく活用されるのが、**広告予算配分の最適化**です。たとえば「どの媒体（メディアに）にどのくらいの金額を出稿（投資）すれば、広告の費用対効果が最大化されるか？」を解くことができます。商品価

格と同様に、出稿金額も連続変数なので、連続最適化に相当します。

また、たとえば金融関連で、どういった銘柄（資産）にどれだけの金額を投資すれば、投資リスク対リターンを最大化できるか？という問題があります。広告の予算配分にもコンセプトが似ていますね。このような金融資産の最適化は、よく「**ポートフォリオ最適化問題**」といわれます。

このように、変数が連続数値となっている場合は、連続最適化という枠組みの中で最適化問題を解くことになります。データサイエンティストやエンジニアでなければ、細かい内部のアルゴリズムはそこまで気にする必要はありませんが、**連続最適化と組み合わせ最適化で、内部でどのようなアルゴリズムを採用して解くかが異なってきます。**したがって、アルゴリズムの細かい部分は考えずとも、そのような実装者と一緒に、ビジネス的にどのような変数を最適化したいか？ そしてそれは連続変数かどうか？ といった視点でディスカッションできると非常によいでしょう。

これは連続最適化に限った話ではありませんが、最適化では、これまでの機械学習のように「過去蓄積した大量のデータを学習させて」というステップは明示的には踏んでいません。極端な話、データがなくとも最適化は解くことができます。 しかし、できるかぎりさまざまなデータを蓄積しておくべきです。たとえば上記のような最適化を解いた結果、求められた最適価格で施策を実施したとしましょう。仮にその結果が芳しくなかったら、そこで得られた結果が「最適化問題の定式をどう改善していこうか」という材料になります。したがって、適切にデータを蓄積することで、最適化結果を施策適用しながら、どんどんと精度を上げていくサイクルを回すことができるようになります。

Section

04 2つの最適化② 「組み合わせ最適化」

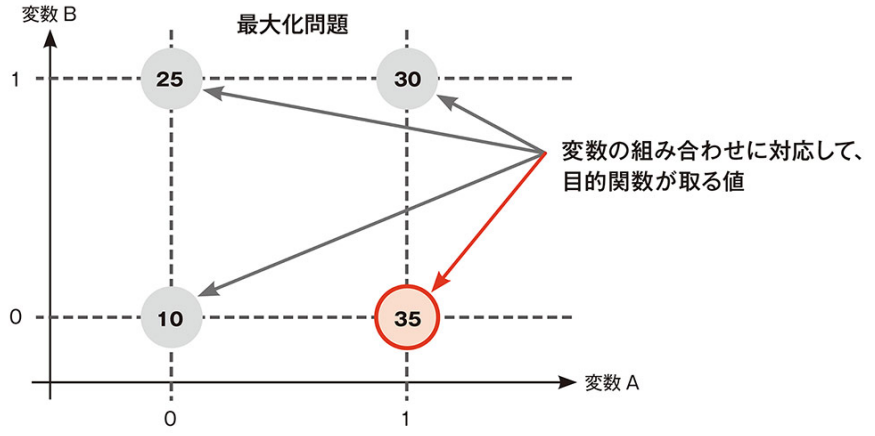
組み合わせ最適化とは？

組み合わせ最適化に関しても紹介しておきましょう。今回の例は連続最適化の問題になりますが、組み合わせ最適化もビジネスで非常によく使われます。組み合わせ最適化とは、**変数が離散的な数値を取るような最適化問題**を指します。離散的な数値とは、[0, 1] といった、取りうる値が飛び飛びになっている数値です。

したがって、たとえば対象となる変数が 2 種類ある場合は、考えられる変数は［図 9-4-1］のような**変数ごとに取る値の組み合わせ**となります。ゆえに組み合わせ最適化と呼ばれているわけです。この組み合わせの中で、目的関数が最も大きい（または小さい）値を取る変数の組み合わせが最適解である、と考えられます。

➲ 組み合わせ最適化の定義［図 9-4-1］

組み合わせ最適化：変数が離散的な数値を取るような最適化問題



この例では、35 が最大値を取るため、最適と判断できる

実務的によく利用される変数としては**「XX するかどうか」という事象を 0 か 1 の変数として利用**するケースです。具体例を見ていきましょう。

ビジネスにおける組み合わせ最適化の事例

連続最適化と同様に、よく使われるビジネス活用例を［図 9-4-2］にいくつか紹介しましょう。

➲ 組み合わせ最適化のビジネス活用例の一例［図 9-4-2］

一例	どのような変数を最適化するか	どのような目的関数を最大 (最小) 化するか
配送ルート最適化	どの車両がどの箇所を訪問するかしないか (0/1)	配送する車両台数 (あるいは移動距離など) を最小化
積付最適化 (ビンパッキング)	どの商品をどの箱に詰めるか詰めないか (0/1)	詰め込む箱数を最小化
シフトスケジュール最適化	その人員をその時間帯に入れるか入れないか (0/1)	稼働する従業員数を最小化

変数や目的関数は一例です。ビジネスケースによって定式化は異なる可能性があります。

1 つは、物流・配送業界などでよく用いられていますが、**ルートの最適化**です。これは、「**ある車両が、ある店舗等の配送先を訪問するならば 1、しなければ 0**」といった変数設定となります。そして、すべての車両・すべての配送先に関する 0/1 変数の組み合わせがわかれば、車両ごとにどういったルートで配送するかがわかります。結果的に、配送する車両台数や、あるいは配送にかかる移動距離などが最小化されていれば、ビジネス上の効果があるといえます。これに似た「**巡回セールスマン問題**」と呼ばれるものがあり、組み合わせ最適化の例としてよく登場します。

2 つ目も同様に 0/1 の変数で考えられる、**積付の最適化**です。これも物流

トラックなどさまざまな場面で活用されますが、「**ある商品を、ある箱（あるいはトラックなど）に 詰め込むならば 1、しなければ 0**」といった変数設定となります。そしてその結果、詰め込む箱数を最小化できれば、トラックなどの使用台数を削減できます。このような詰め込みに関わる有名な問題としては、**「ナップサック問題」や「ビンパッキング問題」**などがあり、よく引き合いに出されます。

3 つ目は**シフトスケジュールの最適化**です。誰をどういった時間帯にシフトに入れるか、という施設責任者などがよく頭を悩ませる問題を、最適化問題として解きます。人員ごとに、シフトに入れない時間帯などの制約条件をクリアしながら、「**どの人をどの時間帯に入れるか（1 か 0 か）**」という変数の最適解を求めることで、稼働人数を最小化することを目指します。

組み合わせ最適化は、その名の通り変数の組み合わせを考えるので、少し勘がいい人は、すべての組み合わせを列挙してしまえばよいのではないかと考えるかもしれません。実はそのやり方でもよいのですが、対象とする商品数や人数などが少なければ成り立ちますが、多くなるとそうもいきません。たとえば前述した巡回セールスマン問題やナップサック問題では、対象の巡回先や商品数が 30 を超えるだけで、数十億を超える組み合わせとなってしまいます。このような現象は「**組み合わせ爆発**」などと呼ばれ、これを回避するために、さまざまなアルゴリズムによって素早く解を求めることを目指します。

余談ですが、最近話題になっている**量子コンピューター**などは、この組み合わせ問題を超高速に解けるといわれており、さまざまな研究が進んでいます。ただし現状ではまだ研究途上な部分も多く、また量子コンピューター自体や取り巻くソフトウェアも社会的に普及が進んでいないので、実務的な最適化問題へ手軽に応用されるのはもう少し先になりそうです。

楽しみな未来が待っているかもしれないという期待がもてます！

Section

05 実践：小売店舗の商品データを活用しよう

練習用ファイル：chap09_price_optimization / optimization.xlsx

実践 1商品における単価と売上個数の関係をモデリング

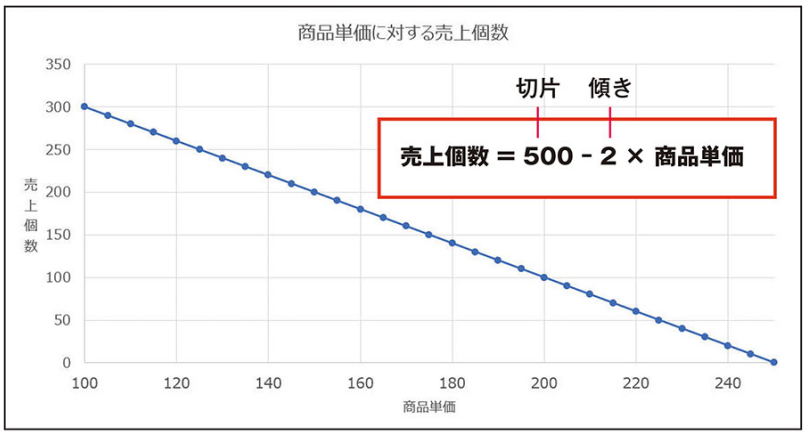
さて、最適化の紹介を終えたところで、冒頭で述べた課題を具体的に解いていきましょう。まずは単一の商品に関して考えた後、複数商品の商品単価を最適化してみます。

1商品における商品単価を最適化するためには、**単価と利益の関係性をモデリング（数式化・定式化）**する必要があります。これらの定式化はとても難しい部分が多いですが、ここでは皆さんに最適化がどういうものかをできるだけわかりやすく伝えるために、とてもシンプルに考えていきます（もちろん実務的にはさまざまな、難しい・詳細な点を考慮する必要があることは頭の片隅に置いておいてください）。

まずは、単価を変えた際に売上個数がどう変化するかを定式化しましょう。事象をとてもシンプルに考え、「安くすればたくさん売れる・高くすればあまり売れない」とします。すると、［図 9-5-1］のように、**商品単価 対 売上個数は、右肩下がりの直線**で表すことができそうです。

「単価と売上個数が負の関係性にある」というのは、単価が上がると需要が下がり売上個数が減るという意味で、いわゆる需要供給曲線と似たような現象として定義している、と考えられるわね。

➲ 単価と売上個数を直線の関係でモデリング［図 9-5-1］



単価を高く（低く）するほど、売上個数が下がる（上がる）ことを表現

ここで、教師あり学習／回帰問題で学んだ線形回帰モデルを導入します。どういった直線式とするかは難しいですが、過去にいくつかの価格で試した情報をもとに、以下の回帰式（直線の式）で表すこととします。

➲ 今回の回帰式［図 9-5-2］

切片を 500、傾きを -2 とし、

売上個数 = 500 - 2 × 商品単価

これにより、商品単価をどう変えると、販売個数がいくつになりそうか？を予想できそうです。［図 9-5-1］は、「chap09_price_optimization」フォルダの「optimization.xlsx」ファイル「1 商品の場合」シートに、商品単価と売上個数をプロットする形で、図示しています。

Tips　実際の制約を念頭にモデリングする

もちろん実際のところは、特に消費財の場合などは、商品単価が上がれば上がるほど、消費者はより買いにくくなるはずです。そのため今回の分析のように直線の仮定をしてしまうと、そのような実態に即さなくなる可能性があります。したがって、より正確にモデリングを行うためには「単価が上がりすぎると売上個数は加速度的に減ってしまう」ということを念頭に置いた商品単価と売上個数の関係式を定義したうえで、最適化するという点も考慮に入れる必要があります。

練習用ファイル：chap09_price_optimization / optimization.xlsx

実践 1商品における単価と利益の関係をモデリング

この前提をもとに、商品単価と利益の関係性をモデリングします。商品単価に対する売上個数が定式化できたので、各単価に対する売上額（＝商品単価×売上個数）がわかります。その試算を同シートのC列「売上額」にて計算しています。またさまざまなコストが存在しますが、仕入れコストのみを加味しましょう。今回は簡単に、以下の仮定をおきます。

➲ 仮定した仕入れコスト［図 9-5-3］

- 仕入先との交渉が難しいということで、仕入単価はすべての商品に関して固定
- （本来は仕入れタイミングや消費期限等も加味する必要があるが、今回は簡略化して）仕入れコストにかかる個数はそのまま売上個数とする

すると、最適化の定式化は次ページの［図 9-5-4］のように定義できます。またその計算を Excel で行うと、D列で仕入れコストの計算をし、C列の売上額からD列の仕入れコストを差し引いた利益をE列に計算できます。

その結果、A列の商品単価とE列の利益をプロットしたものが上に凸な二次関数の形状となっていることがわかります。この二次関数が、そのまま目的関数になっているということです。

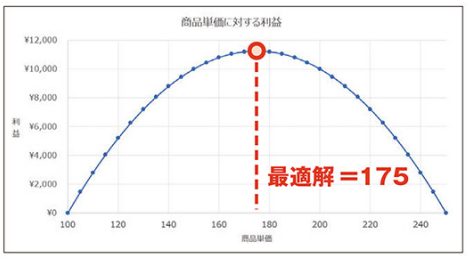
二次関数の形状を解釈すると、単価が下がるとシンプルに売上高が下がるため利益が下がり、一方で単価が上がりすぎると（需要が減り）売上個数が下がるため利益が下がる、と考えられますね。

➲ 商品単価と利益の関係性をモデリング［図 9-5-4］

最適化の定式化

（最大化）目的関数 = 商品の期待利益
= 売上額 - 仕入コスト
=（商品単価 × 売上個数）-（仕入単価 × 売上個数）

商品単価	売上個数	売上額	仕入コスト	利益
100	300	30000	30000	0
105	290	30450	29000	1450
110	280	30800	28000	2800
115	270	31050	27000	4050
120	260	31200	26000	5200
125	250	31250	25000	6250
130	240	31200	24000	7200
135	230	31050	23000	8050
140	220	30800	22000	8800
145	210	30450	21000	9450
150	200	30000	20000	10000
155	190	29450	19000	10450
160	180	28800	18000	10800
165	170		17000	



単価を高く（低く）するほど、売上個数が下がる（上がる）ことを表現

このことから、利益が最大となるような商品単価の最適解は 175 であることがわかりました！ もちろん現実的には以下のような点も考慮に入れないといけないため、今回ほど簡単には計算できないでしょう。

- そもそも商品単価と売上個数は、直線の関係ではない（曲線などの）可能性がある
- 売上個数に影響を及ぼしているのは商品単価のみではなく、天気や曜日特性などさまざまな変数も考慮に入れる必要ある

したがって、実務的にはより複雑な最適化問題を解く必要があり、今回のように解析的に[※4]求めることはできないケースも多いです。ただし、**現象をモデリング（定式化）し、変数と目的関数を定義し、最適解を計算する**、というイメージはついたのではないでしょうか。

（※ 4）四則演算や数学の「微分」などの知識を用いて、数式の変形だけで最適解を求めることを指します。

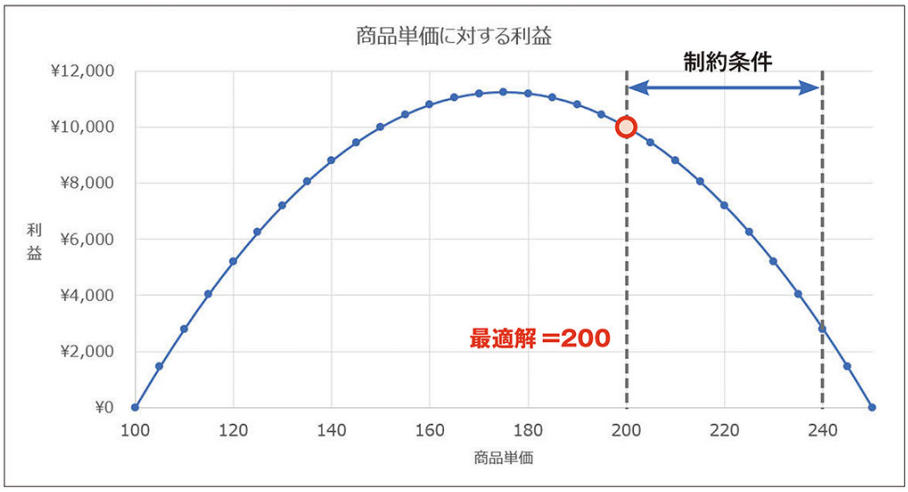
練習用ファイル：chap09_price_optimization / optimization.xlsx

実践 制約条件がある場合

もし制約条件がある場合は、それも加味する必要があります。実務上はさまざまな制約があるはずです。たとえば「商品単価は仕入れ単価より高くする」「高すぎると買われないからXX円未満とする」といったようなイメージです。

そういった**制約条件を満たしたうえで、目的関数が最良な値を最適解**とします。たとえば［図9-5-5］のような制約条件（商品単価は200円以上240円以下）があったならば、その中で最良な状態、つまり200円が最適解である、といったように判断することができます。

➲ 制約条件がある場合のイメージ［図9-5-5］



制約条件があれば、それも加味した最適解を算出する

練習用ファイル：chap09_price_optimization / optimization.xlsx

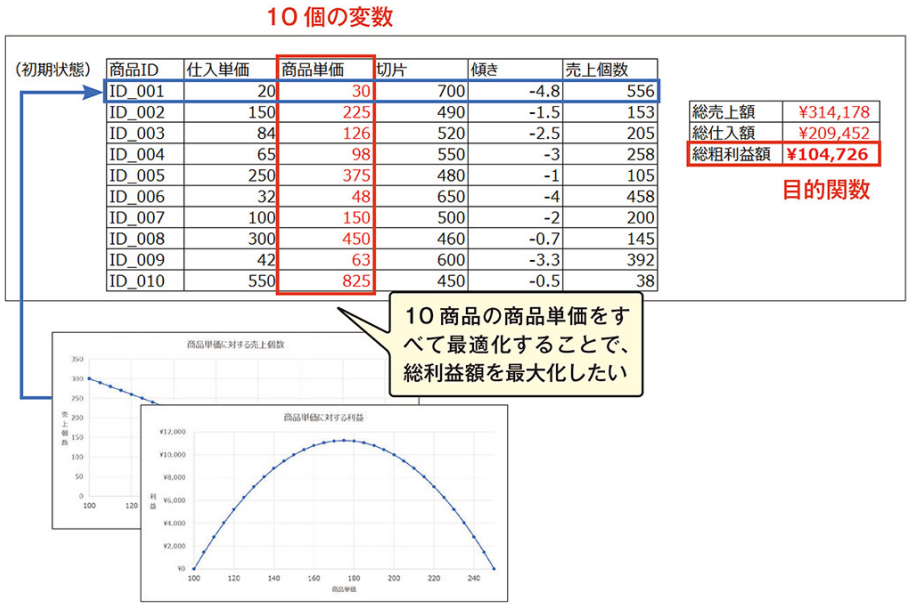
実践 複数商品における利益を最大化したい

せっかくなので、もう少し発展的な分析をしてみましょう。実際には1つの商品だけではなく、複数の商品を取り扱っているはずです。そこで、これまでの最適化を複数商品同時に最適化してみましょう。「複数商品の場合_最適化を実行」シートに、10商品の情報があります。

各商品の仕入単価および商品単価の初期値（現在値）をC～D列に記載しています。また商品ごとに単価と売上個数の関係性が異なるため、それぞれの商品で過去の情報や傾向をもとに、直線式を表すための切片と傾きをE～F列に記載しています。すると、1商品の場合と同様に商品ごとの売上個数が求められ、全商品での総売上額・総仕入額から総利益額が計算できます。今回対象とする10商品の期待される利益額は、現状約100,000円であることが見てとれます。

この10商品の商品単価を動かし最適化することで、全商品の総利益額を最大化することを目指しましょう。

➲ 複数商品の商品単価を同時に最適化する［図 9-5-6］



前述したように、今回の最適化問題は非常にシンプルに定式化しています。先ほど試したような1商品ごとの単価と利益の関係性を図示すれば求められますが、10商品すべて計算・可視化するのは少々面倒です。

実務的には最適化問題を解くための問題設定が複雑なことがよくあります。そのため先ほどのように可視化して算出することはあまりせず、何かしらの計算ツールを使用することが多いのです。計算ツールの候補としては、以下のものが考えられます。

➲ **計算ツールの例**［図 9-5-7］

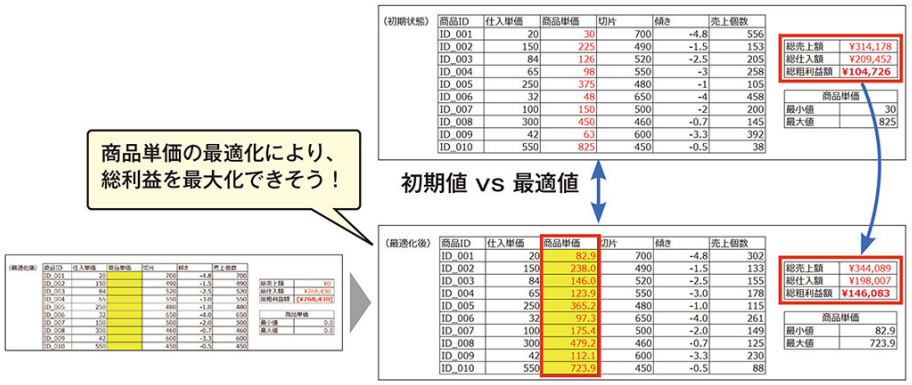
- **最適化問題が比較的小規模・シンプルな場合**
  Excelの「ソルバー」機能
- **最適化問題が比較的大規模・複雑な場合**
  Pythonなどのプログラミング言語

今回はシンプルな問題設定なので、Excelのソルバー機能で最適化問題を解いてみました。ソルバー機能の使い方については本書では解説しないので、Webなどで調べてみてください。

最適化結果は［図9-5-8］右下の通りです。Excelでは、L列以降に記載しており、O列に最適化された商品単価を記載しています。初期状態と比較すると、総利益額が約146,000円となっており、約100,000円の状態から改善しています。つまり、**初期状態の変数の値を、最適化により最適解にした**ことで、**目的関数である総利益額を最大化**できている、ということです。

「最適化問題が複雑になる」とは、たとえば、そもそも単価と売上個数の関係が線形（直線）ではなかったり、天候や曜日などほかにもさまざまな変数を考慮しなければいけなかったり、あるいはより複雑な制約条件が加わってきたりする、といったケースが該当するわね。この場合は、Excelだけでは対応が難しくなるため、プログラミング言語による実装が必要になるでしょう。

➲ 各商品単価の最適化により、総利益を最大化する［図 9-5-8］



最適化による計算後

複数商品の場合も、今回のようなシンプルな問題設定であれば、解析的に解けてしまいますが、実際にはより複雑なモデリングや制約条件が加わることとなります。

たとえば、仮にある商品を値下げしたら、ほかの似た商品を買おうと思っていたユーザーは、値下げされた商品を買おうとするインセンティブが働くかもしれません。その場合は、複数商品における価格決定が販売個数に影響を及ぼすような問題設定を考えなければなりません。

そうなるとかなり数学的に複雑な定式化になってしまうので、今回はできるだけ数式を排除した形で最適化を紹介しましたが、その概念や考え方自体はそこまで難しくなかったはずです。数理最適化の考え方は、実社会やビジネスでも思考を整理するためのフレームワークとしても活用できるはずであり、最近ではコンピューティングリソースも潤沢になってきているため、実務での活用事例も増えてきている印象があります。この機会に理解を深め、皆さんの身の回りやご自身の業務で数理最適化で解けそうな問題がないか、ぜひ考えてみてください。

■ ここで学んだ重要トピック

- 最適化（変数、目的関数、制約条件）
- 連続最適化
  （ダイナミックプライシング、広告予算配分最適化、ポートフォリオ最適化など）
- 組み合わせ最適化
  （巡回セールスマン問題、ナップサック問題、ビンパッキング問題、シフトスケジューリングなど）
- 組み合わせ爆発
- Excel ソルバー

■ ステップアップにつながるトピック

- 線形計画問題、整数計画問題、2 次計画問題 など
- NP 困難
- 勾配降下法、ニュートン法、内点法 など
- メタヒューリスティック解法（遺伝的アルゴリズム、焼きなまし法 など）

さて、ここまででデータサイエンスの基礎の伝授は終了よ。このあとのページに、ステップアップにつながるオススメ書籍を紹介しておいたので、より深く学ぶのなら参考にするといいわ。

ここまでが基礎なんですか。まだまだ先は長いんですね……。

そう。先は長いわよ。でもそれは可能性が無限にあるということ。データサイエンスはこの先ずっと使えるし、必ず役立つスキルだから、ここで得た知識を応用してさまざまな課題を解決していきましょう！

おわりに

本書を最後まで読み進めていただき、どうもありがとうございました。ここまで、データサイエンスにおける基本的かつ重要な技術要素を、ビジネスケースをもとに解説してきました。データサイエンスの技術を、できるだけ数式を使わずに紐解いて説明し、かつ、それをビジネスの施策適用につなげていく具体的なイメージが読者の皆さんに湧くように意識して執筆しました。もちろん、実務では本書のケースのように簡単にはいかないことも多いでしょう。そのためまだまだ書き足りないこともありますが、まず手始めに押さえておきたいポイントを紙幅の許す限り盛り込んだつもりです。

本書の内容に関して、新しい発見が多ければ幸いですが、理解が難しかった部分もあると思います。しかし、一度に本書の内容をすべて理解する必要はありません。私自身も、最初にデータサイエンスに関する分野を学び始めた際は、理解が及ばないことも数多くありました。そして今でも道半ばで、まだまだ学ばなければならないなと感じることが非常に多いです。ですので、読者の皆さんも、もしかしたら「やっぱりAI・データサイエンスは難しいな……」と感じた方もいるかもしれませんが、わからない部分があれば何度も読み返してみてください。また、そのほかの本やWeb上の情報を収集し、理解を深めていくくこともおすすめします。

また、もし「本書ではちょっと物足りないな」と感じたら、ぜひさらなる学びにチャレンジしてみてください。今回紹介した教師あり学習やレコメンデーションといった技術に関して、より難しいトピックを学ぶもよし、または今回は実装自体はしていませんが、プログラミング言語であるPythonなどによる実装・開発を学ぶのもよいと思います。

AI・データサイエンスという世界は、もしかしたら響きは綺麗に聞こえるかもしれません。しかし私自身、実務上なかなか難しい領域であると痛感しています。当然ビジネスにおいては、データ活用だけでは解決できない課題も多いです。また何より技術自体が難しい分野ということもあり、周囲のメンバーや部署、会社から理解を得て進めることが比較的大変であると感じ

ます（私の実力不足な部分も多分にあると思いますが……）。そういった課題感も、今回執筆にいたった背景にあります。本書を読み、少しでもAI・データサイエンスに興味を持ったり、身近に感じたり、あるいは自社などで何か実践できないか？と思ったりするきっかけが増えれば、これほどうれしいことはありません。

ここ近年のめざましい成長を遂げている企業では、ほぼ例外なくAI・データサイエンスの活用が進んでいます。GAFAMのようなビッグテック企業はもちろんのこと、米国のNetflix、Airbnb、中国のBaidu、Alibaba、Tencentなどの企業も、AI・データサイエンスに関するビジネス適用が非常に盛んです。またそれらの企業はAI研究所といったR&Dセンターも設置し、中長期における投資もしっかりと進んでいます。またIT企業だけではなく、米国ウォルマートや中国Alibabaのフーマーフレッシュなど、非IT領域に主軸をおくような企業やサービスにおいても、AI・データサイエンスの活用を積極的に進めている企業が見受けられます。私たちもそういった企業の成功事例を見ながら、ビジネス、ひいては社会をよりよくしていくために、一緒にAI・データサイエンスを学び、活用していければうれしいです。

私自身、企業様中心に、さまざまな方々にAI・データサイエンスに関わるコンサルティングや教育、ソリューション開発などの支援をしていますので、もし実務での新たなチャレンジとしてデータ活用の取り組みを考えられそうな方は、いつでも気軽にお問い合わせ・ご相談ください。ぜひ、いろいろとディスカッションしましょう。

2022年春　三好大悟

ステップアップにつながるトピックまとめ

第3章から第9章の章末に紹介した「ステップアップにつながるトピック」を簡単に解説します。

Chapter 3

● ピボットテーブル

表計算機能の1つで、表形式のデータから、各カテゴリごとに変数の統計量の差異を集計する機能（例. 店舗A：売上高X円、店舗B：売上高Y円…）

● 確率分布（正規分布、二項分布、ポアソン分布 など）

現実に起こる事象を確率で表現する数式。事象の性質によって正規分布、二項分布、ポアソン分布などさまざまな確率分布が存在する

● 仮説検定（t検定、カイ二乗検定 など）

ある仮説が正しいか否かを、手元のサンプルデータ及び確率分布を用いて、統計的に検証する方法論。代表的なものにt検定、カイ2乗検定などがある

● データの前処理（ダミー変数、欠損値の補完、外れ値の考慮 など）

現実のデータに欠損値や外れ値があった場合に必要な処理のこと。ダミー変数を作成したり、欠損値を補完したりなどデータの前処理を行ってから線形回帰モデルなどのモデリングを行う

Chapter 4

● 決定木

線形回帰モデルと同じく教師あり学習の手法の1つ。データの構造を線形ではなく木構造で表現するモデル

● ランダムフォレスト（XGBooxt、LightGBM）

決定木のモデルを多数用意し、学習・統合させる複雑なモデル。線形回帰モデルや決定木と比較すると、予測精度が高く出るために有用される。ランダムフォレス

トを発展させた類似のモデルにXGBoostやLightGBMなどがある

● ハイパーパラメータ

機械学習モデルの学習プロセスを制御する変数。決定木において、どのくらい木の構造（深さ）を複雑にするか、など

● 過学習／未学習

手元のデータに対して学習しすぎる／し足りない状態で、未知のデータに対する予測精度が高くならない現象

● 汎化性能

未知のデータに対する精度。汎化性能が高い＝過学習も未学習もしていなく、未知のデータへの精度が高い状態

● クロスバリデーション

手元のデータで、作成したモデルの汎化性能を検証する方法論

● グリッドサーチ、ランダムサーチ、ベイズ最適化

複数存在するハイパーパラメータを探索する方法論。いくつかの方法が存在し、クロスバリデーションと一緒に用いられることが多い

Chapter 5

● 交差エントロピー誤差関数

ロジスティック回帰モデルにおける目的関数

● 尤度、対数尤度

データが規定した確率分布などにどの程度フィッティングしているかを示す指標。交差エントロピー誤差関数の定義にも使用される

● ニュートン法、最急降下法、勾配降下法

交差エントロピー誤差関数を最小化するために、ロジスティック回帰モデルのパラメータを最適化するための方法論。いくつかの方法論が存在する

●一般化線形モデル

線形回帰モデルを発展させた、線型性を有するモデル群。ロジスティック回帰モデルも一般化線形モデルの一種で、ほかにもいくつかのモデルが存在する

●決定木、ランダムフォレストなどによる分類問題

決定木やランダムフォレストなどの木系のモデルは回帰問題だけではなく、分類問題にも対応している

●AUC

分類問題のモデルの精度を計測する指標の1つ。特に（予測フラグではなく）予測確率を直接的に評価する際に使用できる

●F-betaスコア

F1スコアの発展形。PrecisionとRecallに重みを付けた平均値で評価する指標。どちらにどの程度重みをつけるかを指定することができる

●不均衡データへの対応アプローチ

分類問題における目的変数0/1のデータ数が極端に不均衡な（特に1のデータ数が少ないことが多い）場合では精度が下がるケースがあり、その際に精度を上げる方法論。代表的なものにOver / Under sampling、SMOTEなどがある

Chapter 6

●勾配降下法、確率的勾配降下法、AdaGrad、Adam など

規定した目的関数を最小化するために、重みパラメータを更新していき、最適化する方法論

●誤差逆伝播法

複数層ある深いネットワーク構造の際に、入力層から出力層にかけて、すべての層の重みパラメータを適切に学習させるためのアルゴリズム。英名Back Propagtation

●活性化関数

ネットワークの各ノードにおいて、出力値を決定するための関数。どれを選ぶかで精度が変わってくることが多い。活性化関数の一例としてシグモイド関数、

ReLU関数などがある

● Softmax関数

ロジスティック関数を多次元に拡張した関数。多クラス分類問題において、多クラスの予測確率を出力するためにネットワークの最後に用いられることが多い

● Stride、Kernel、Padding など

畳み込みネットワークにおけるさまざまなハイパーパラメータ

● Dropout、Batch Normalization

ネットワークを過学習させないための方法論

● Attention、Transformer、BERT

近年発展している、より複雑なネットワーク。大量のデータを大量のパラメータで学習させている

● 発展的な画像解析手法

画像分類以外の発展的な画像解析手法。物体検出、物体追跡、姿勢推定など。Chapter2でいくつか紹介している

Chapter 7

● 初期値依存性の解消、k-means++法 など

K-means法におけるアルゴリズムを改善するための方法論

● エルボーメソッド、シルエットプロット など

最適なクラスタ数を決定するための理論的な方法。クラスタの解釈と併用し、最適なクラスタ数を考慮することが多い

● 階層型クラスタリング、スペクトラルクラスタリング

ときにK-means法では捉えられない、複雑なクラスタリングができるような、その他の教師なし学習の手法

● 次元圧縮（PCA、SVD、t-SNE など）

次元数（変数の数）が大量にある場合に、それを削減する方法論

Chapter 8

●ジャッカード係数、ユークリッド距離 など

コサイン類似度以外にもいくつかの類似度計算方法が存在し、データを見ながら類似度を定義していくことが多い

●オフライン検証（MAP、nDCG など）

ABテストのようなオンラインテストが難しい場合には、オフラインでいくつかの評価指標を用いた机上検証を行う

●教師あり学習を活用したレコメンド

購買履歴などを学習データとし、教師あり学習モデルを学習し、レコメンドに活用することもできる。近年ではディープラーニングを用いた手法も増えてきている

Chapter 9

●線形計画問題、整数計画問題、2次計画問題 など

定式化された最適化問題の種類。最適化問題を解く際には、どのような種類かを意識する必要があることが多い

●NP困難

最適化問題を解くための計算量が非常に大きいことを指す

●勾配降下法、ニュートン法、内点法 など

最適化問題を解くための方法論。教師あり学習におけるパラメータを最適化するためにも用いられる

●メタヒューリスティック解法（遺伝的アルゴリズム、焼きなまし法 など）

特定の計算問題に依存しないで最適化問題を解く方法論。NP困難なアルゴリズムに対して用いられることが多い。代表的なものとして遺伝的アルゴリズム、焼きなまし法がある

ステップアップにつながる書籍

→ 統計学に関して、より理解や学びを深めたい方へ

統計学の理論をちゃんと学ぶ

『統計学入門（基礎統計学Ⅰ）』
東京大学教養学部統計学教室（編集）、東京大学出版会（刊）

プログラミング言語 R で統計学を学ぶ

『R によるやさしい統計学』
山田 剛史（著）、杉澤 武俊（著）、村井 潤一郎（著）、オーム社（刊）

→ データサイエンス全般に関しての学びを深めたい方へ

データサイエンスをビジネスに活かす

『戦略的データサイエンス入門
——ビジネスに活かすコンセプトとテクニック』
Foster Provost（著）、Tom Fawcett（著）、竹田 正和ほか（訳）、
オライリージャパン（刊）

プログラミング言語 Python で データサイエンスを学ぶ

『東京大学のデータサイエンティスト育成講座
～ Python で手を動かして学ぶデータ分析～』
中山浩太郎（監修）、塚本邦尊ほか（著）、マイナビ出版（刊）

→ 機械学習に関して学んでみたい方へ

機械学習をビジネスで使う

『仕事ではじめる機械学習 第 2 版』
有賀 康顕（著）、中山 心太（著）、西林 孝（著）、オライリージャパン（刊）

Python を使って機械学習を本格的に学ぶ

『Python ではじめる機械学習
—— scikit-learn で学ぶ特徴量エンジニアリングと機械学習の基礎』
Andreas C. Muller（著）、Sarah Guido（著）、中田 秀基（翻訳）、
オライリージャパン（刊）

Index

数字・アルファベット

あ

か

さ

た

な

は

ま

や

ら

◉ 著者プロフィール

三好 大悟（みよし・だいご）

慶應義塾大学理工学部で金融工学を専攻。大学卒業後、株式会社データミックスにてデータサイエンティストとして、統計学や機械学習を用いたデータ分析・アルゴリズム開発を中心としたコンサルティングに従事。2020年7月からは株式会社セブン&アイ・ホールディングスにて、小売や物流・配送などの事業におけるデータ・AI活用を推進。一方で兼業としても活動し、データ分析やAI開発など、データサイエンスに関するアドバイザリ・受託開発・教育活動などにも携わる。
daigo.miyoshi@liber-craft.co.jp

◉ STAFF

カバー・本文デザイン　株式会社Isshiki
カバー・本文イラスト　北川ともあき
DTP・図版作成　西嶋 正

デザイン制作室　今津幸弘
制作担当デスク　柏倉真理子

テクニカルレビュー　佐藤 圭

編集協力　浦上諒子
副編集長　田淵 豪
編集長　藤井貴志

■商品に関する問い合わせ先
このたびは弊社商品をご購入いただきありがとうございます。本書の内容などに関するお問い合わせは、下記のURLまたはQRコードにある問い合わせフォームからお送りください。

https://book.impress.co.jp/info/

上記フォームがご利用頂けない場合のメールでの問い合わせ先
info@impress.co.jp

※お問い合わせの際は、書名、ISBN、お名前、お電話番号、メールアドレスに加えて、「該当するページ」と「具体的なご質問内容」「お使いの動作環境」を必ずご明記ください。なお、本書の範囲を超えるご質問にはお答えできないのでご了承ください。

●電話やFAXでのご質問には対応しておりません。また、封書でのお問い合わせは回答までに日数をいただく場合があります。あらかじめご了承ください。
●インプレスブックスの本書情報ページ https://book.impress.co.jp/books/1121101015 では、本書のサポート情報や正誤表・訂正情報などを提供しています。あわせてご確認ください。
●本書の奥付に記載されている初版発行日から3年が経過した場合、もしくは本書で紹介している製品やサービスについて提供会社によるサポートが終了した場合はご質問にお答えできない場合があります。

■落丁・乱丁本などの問い合わせ先
TEL 03-6837-5016 FAX 03-6837-5023
service@impress.co.jp
（受付時間／10:00〜12:00、13:00〜17:30土日祝祭日を除く）
※古書店で購入された商品はお取り替えできません。

■書店／販売会社からのご注文窓口
株式会社インプレス 受注センター
TEL 048-449-8040
FAX 048-449-8041

ビジネスの現場（げんば）で使（つか）える AI（エーアイ）＆データサイエンス（アンド）の全知識（ぜんちしき）（できるビジネス）

2022年3月11日 初版発行

著 者 三好大悟（みよしだいご）
発行人 小川 亨
編集人 高橋隆志
発行所 株式会社インプレス
〒101-0051 東京都千代田区神田神保町一丁目105番地
ホームページ https://book.impress.co.jp/
印刷所 音羽印刷株式会社

ISBN978-4-295-01363-1 C3055
Printed in Japan