# 保険業界におけるビッグデータ活用に関する考察

日本アクチュアリー会 IT 研究会第３グループ

【担当委員】

| | |
|---|---|
| ﾄｰﾏﾂ | 長瀬　正憲 |
| ﾆｯｾｲ情報 | 高瀬　有香 |

【メンバー】

| | | | |
|---|---|---|---|
| あいおいﾆｯｾｲ同和 | 加藤　健太 | ｱﾌﾗｯｸ | 戸田　祐樹 |
| ｱﾌﾗｯｸ | 塩野　朔也 | かんぽ生命 | 京野　英貴 |
| ｼﾞﾌﾞﾗﾙﾀ生命 | 永久保彩花 | 住友生命 | 瀬尾　拓哉 |
| 第一生命情報ｼｽﾃﾑ | 清水　裕輝 | 第一生命情報ｼｽﾃﾑ | 谷澤　慶典 |
| 大同生命 | 小西　直樹 | ﾆｯｾｲ情報 | 臼井　孝典 |
| ﾆｯｾｲ情報 | 矢口　太朗 | 富国生命 | 入倉　真樹 |
| ﾒｯﾄﾗｲﾌ生命 | 佐久間敏彦 | ﾒｯﾄﾗｲﾌ生命 | 佐藤　智行 |
| ﾒｯﾄﾗｲﾌ生命 | 和久田訓寛 | PGF 生命 | 原田　博行 |
| RGA | 笹生　健 | | |


目次

# はじめに

昨今、政府の進める健康・医療戦略でのレセプト情報公開や自動運転技術の普及などにより多様なデータ蓄積基盤が実現してきている。また、データ分析の分野においてもAIの導入をはじめ、技術は着実に進化しており、ビッグデータの活用事例も急速に増加してきている。

一方で保険業界に目を向けると、一部の商品開発等にビッグデータを活用した事例はあるものの、他業界と比較するとビッグデータ活用はあまり進んでいないように見受けられる。そこで我々第３グループでは、保険業界がビッグデータ活用を進める上で何らかの阻害要因があると考え、生命保険会社・損害保険会社・その他共済団体向けにアンケート調査を実施した。本論文では各社へのアンケート結果や他業界・海外保険業界での好事例より保険業界がビッグデータ活用を進める上で解決すべき課題を考察し、保険業界におけるビッグデータの更なる活用に向けた提言を行う。

# 第Ⅰ章　国内保険会社におけるビッグデータ活用の現状

## Ⅰ－１　ビッグデータとは

ビッグデータというワードは、2010 年初頭から使われ始め、従来の IT システムなどでは記録や保管、解析が難しい巨大なデータ群に対して用いられてきた。その対象として企業の顧客情報から、SNS 上のキーワード、天気、GPS、政府統計まで多様な情報が挙げられ、また AI による解析と組み合わせることで新たな価値を創造することができることから、ビジネス的な利用価値の大きさに注目が集まっている。

本論文で扱うビッグデータの定義は、一般に用いられるものを踏襲し、“Volume（容量の大きさ）・Variety（多様性）・Velocity（発生頻度の高さ）のすべてあるいはいずれかの要素を含むデータの総称”とする。

ビッグデータの大きな特徴の一つとして、非構造化、非定型データも含まれていることが挙げられる。一般にデータは行と列で構成された表形式で管理されることが多いが、SNS でのメッセージ、画像、文書など、特定の構造を持たず意味づけがされていないデータを非構造化データまたは非定型データと呼ぶ。電子機器の普及、モバイルでのネットの利用の増加に伴い、非構造化データを生成するユーザ・デバイスの数は現在でも飛躍的に増加を続けている。従来のデータ処理では、データに意味づけされた情報をもとに、検索、集計を行うものが一般的である。それゆえに非構造データは解析が難しく、各業界でのビッグデータの導入を阻む一因ともされた。次節では、我々が行ったアンケート結果をもとに現在の国内保険業界でのビッグデータの活用状況について言及し、活用事例について整理を行う。

## Ⅰ－２　国内保険会社のビッグデータ活用状況

総務省平成 29 年版情報通信白書[1]では、“2016 年末から 2017 年にかけて、官民データ活用推進基本法の制定や改正個人情報保護法の全面施行などといった法整備が進められた”ことで、2017 年に日本でもようやく「ビッグデータ利活用元年」を迎えたとの報告がされている。2018 年は、これまでにも増して新聞紙面・各メディアニュースから AI・ビッグデータの関連記事が取り上げられた一年となった。そのような背景のもと、各保険会社でもビッグデータの活用に本格的に取り組み始めたのが 2018 年であったと考える。そこで我々は国内の保険会社におけるビッグデータ活用の現状を把握するため、2018 年 8 月に各社アンケートを行い、34 社より回答を受領した。

「AI・ビッグデータの業務への活用を検討しているか？」という質問には、８割弱の会社が検討していると回答したのに対し、「AI・ビッグデータを業務へ活用できているか？」という質問に対しては、活用できているという回答は３割にとどまった。

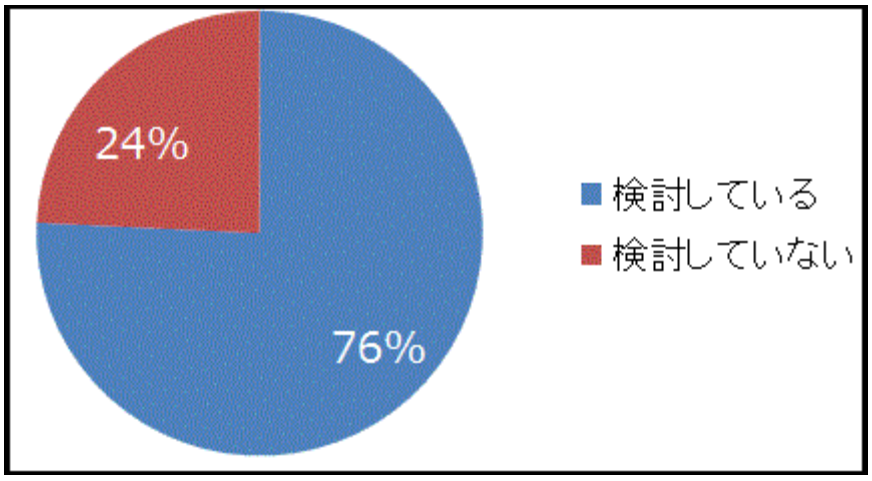
また少し前のデータではあるが、前述の情報通信白書によると、国内の一般企業の約半

[1]http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/pdf/index.html

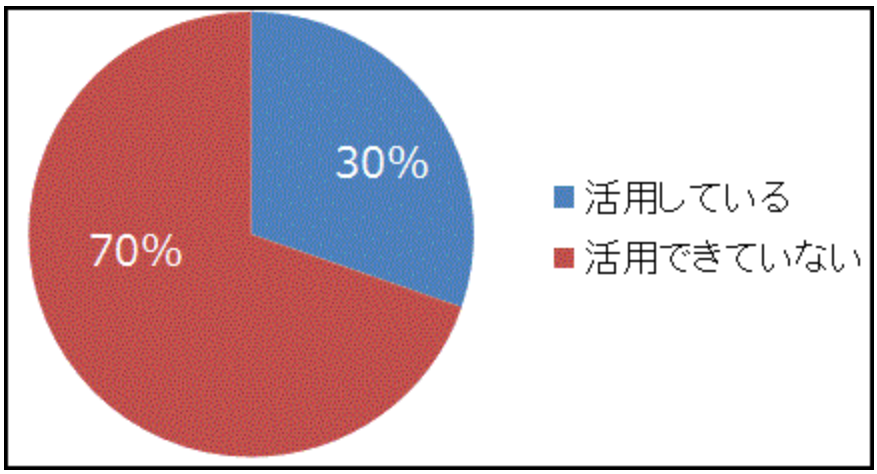
数が「産業データ」を「既に積極的に活用している」あるいは「ある程度活用している」と答えている。

これらの結果より、ビッグデータ活用の導入に向けて国内保険会社の多くは既に（検討を含め）何らかの動きを始めている一方、他業界に比べ、実際の業務に適用する段階までにはまだ到達できていないケースが多いという現状が浮かび上がってきた。

図Ⅰ－１：AI・ビッグデータを業務へ活用を検討しているか




図Ⅰ－２：AI・ビッグデータを業務へ活用をできているか




### Ⅰ－３　国内保険会社のビッグデータ活用事例

各保険会社がニュースリリースしている情報からビッグデータの活用取り組みを抜粋すると、現在日本におけるビッグデータの活用は大きく３つのパターンに分けられる。

１点目は、健康診断など契約者の健康情報をもとにサービスを提供しているパターンである。例えば、生命保険ではネオファースト生命や住友生命、損害保険ではあいおいニッセイ同和損害保険や損害保険ジャパン日本興亜が、ビッグデータ分析の結果を直接保険料等へ反映される保険商品を提供している。２点目は、契約者の健康情報をスマートフォンアプリやウェアラブル端末から収集し、契約者（またはアプリ利用者）の健康サポートを

行うパターンである。３点目は営業職員への支援として契約者情報から特定保険商品の成約予測を行うパターンである。

表Ⅰ－１：国内保険会社におけるビッグデータ活用事例

| データ種別 | 企業 | 対応 | 参考 |
|---|---|---|---|
| 契約者健康情報の活用 | ネオファースト生命<br>住友生命 | 健康診断結果などから“健康年齢”で保険料を決定 | ネオ de 健康エール<br><br>Vitality |
| | アクサ生命 | 健康状態の判定アプリにて健康習慣改善を提案 | Health U |
| | 日本生命 | アプリ連携で健康サポートマイルを発行（予定） | aruku&（あるくと） |
| | 明治安田生命 | ダイエット支援アプリ、中小企業向け健康経営の推進 | Finc パーソナルコーチ AI |
| 契約者運転データ | あいおいニッセイ同和損害保険 | 運転挙動反映型テレマティクス＋ボーナスマイル付与 | |
| | 損害保険ジャパン日本興亜 | 運転診断結果テレマティクス | スマートフォンアプリの活用 |
| 医療患者データ | 朝日生命 | 介護保険商品の成約予測 | データマイニング |

### Ⅰ－４　国内保険会社における活用データ種類

次に、保険会社が利用しているデータの傾向について考える。これまで保険会社では、保険の加入に必要なデータならびに保全・収納事務を中心とした各種保険事務を行うためのデータを、契約者情報をマスターファイルとして管理するにとどまっていた。しかし近年になって、外部の医療情報をビッグデータとして分析し活用を始めた企業も出てきている。例えば第一生命保険では日立製作所との共同研究で「生活習慣病に起因する入院の可能性とその日数」を予測する定量評価モデルを開発し、健康状態を理由に保険加入が難しかった顧客も、保険の引受け基準の見直しにより、新たに加入可能になったという事例もある。しかしながら、ここでも保険業界との関係性の強い医療業界のデータ活用であり、これまで無関係と思われていた社外データとの組み合わせにまでは至っていない。

表Ⅰ－２：国内保険会社における活用データ種類

| | 社内データ | | | 社外データ | |
|---|---|---|---|---|---|
| | 既契約情報 | 医療・ヘルスケア情報 | | 医療情報（レセプト等） | その他 |
| | | 加入時データ（健康診断結果等） | 契約中データ（ライフログ等） | | |
| 商品開発 | ○ | ▲一部 | ▲一部 | ▲一部 | ― |
| 提案書作成<br>※リコメンド | ○ | ― | ― | ― | ― |
| 販売支援 | ○ | ― | ― | ― | ― |
| 引受・加入査定 | ○ | ▲一部 | ― | ▲一部 | ― |
| 支払査定 | ○ | ― | ― | ▲一部 | ― |
| その他契約事務 | ○ | ― | ― | ― | ― |
| 契約者満足 | ― | ― | ▲一部 | ▲一部 | ― |

# 第Ⅱ章　他業界・海外保険業界におけるビッグデータ活用状況

## Ⅱ－１　他業界・海外保険業界におけるビッグデータ活用の現状

他業界や海外保険業界に目を向けると、ビッグデータ活用は進んでおり、国内保険業界においても活用のヒントとなるものがあるのではないかと考えた。本章では、他業界・海外保険業界でのビッグデータ活用事例を紹介し、特徴および優れた点から、保険会社に取り入れるべき要素がないかを考察していく。

## Ⅱ－２　社内他部署とビッグデータ専門部署のコラボレーション事例（大阪ガス　ビジネスアナリシスセンター）

大阪ガスでは、一般企業の IT 部門に相当する情報通信部の中に、９人の分析エキスパートで構成される「ビジネスアナリシスセンター」を有している。

（１）特徴

ビジネスアナリシスセンターは当初、必要なガス消費量の予測のため設置されたが、分析ノウハウの蓄積により活動範囲を広げている。

現在のこの組織のミッションは「企業の全組織・全業務・全サービスにおいてデータ分析の活用機会を発掘し、分析力で新たな価値を創造する」ことである。

ビジネスアナリシスセンターは独立採算制を採用しており、関連会社を含めた大阪ガスグループ内の全組織に１年間で 100 件近いソリューションを提供している。

分析案件ごとに人件費も含め、クライアントとなった組織に費用を請求し、運営している点が特徴である。

（２）実績

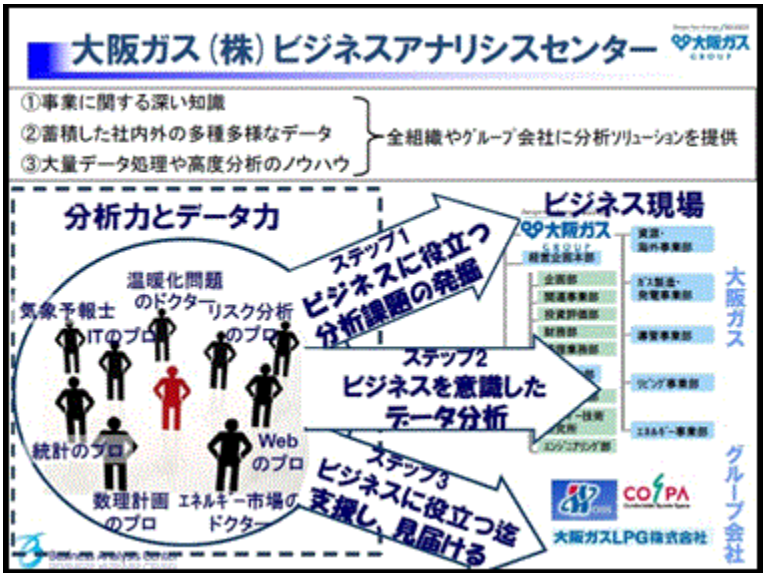
「ビジネスアナリシスセンター」がこれまでに実施したデータ分析業務の実績の例としては、以下が挙げられる。

・緊急車両出勤体制の効率化支援（過去の出勤データや交通渋滞データを分析）
・ガス給湯器の顧客からの問い合わせ実績を利用した分析結果から、給湯器の予防メンテナンス計画を最適化（分析結果の活用で保守員の再訪問回数を抑制でき、顧客満足度を向上）

（３）展望

ビジネスアナリシスセンターは大阪ガスグループ内のデータ分析業務を請け負う組織であるが、近年はその知名度を外部に広め、「データ分析力」を「ブランド」として企業価値を高めていくため、講演会・メディアでの広報・宣伝活動に取り組んでいる。

図Ⅱ－１：大阪ガス（株）ビジネスアナリシスセンターの取り組み[2]




## Ⅱ－３　関係グループ内でのデータ共有事例（中国　平安保険グループ）

海外の保険業界においては、中国の平安保険グループがビッグデータを活用している企業として挙げられる。平安保険グループは 2025 年までの目標として「IT×金融×生活サービスの融合」を戦略に掲げ、世界的な総合金融機関として、ネットを通じてユーザーの生活に密着したサービスを提供する最大手企業グループを目指している。

（１）特徴

中国の大手保険会社の多くが国有企業であるなか、平安保険は民間の保険会社である。平安保険ではこれまで国による実験的な施策や新たな措置を率先して導入してきた。

平安保険の事業の柱は従来は「保険」「銀行」「投資」であったが、最近では「フィンテック」を新たな柱として「インターネット＋総合金融」の戦略を打ち出している。

（２）実績

平安保険グループの保険商品など金融商品の個人顧客およびネット利用を含めた同社のサービスを利用する顧客の総数は約４億人に達している。これら個人の属性に関する４億人分の情報がビッグデータとして平安保険に集まっている。近年ではネット金融の成長が著しく、ネット経由で保険・銀行といった既存の金融商品を購入する顧客が増えており、確実に「重ね売り（クロスセル）」による相乗効果を得ることができている。

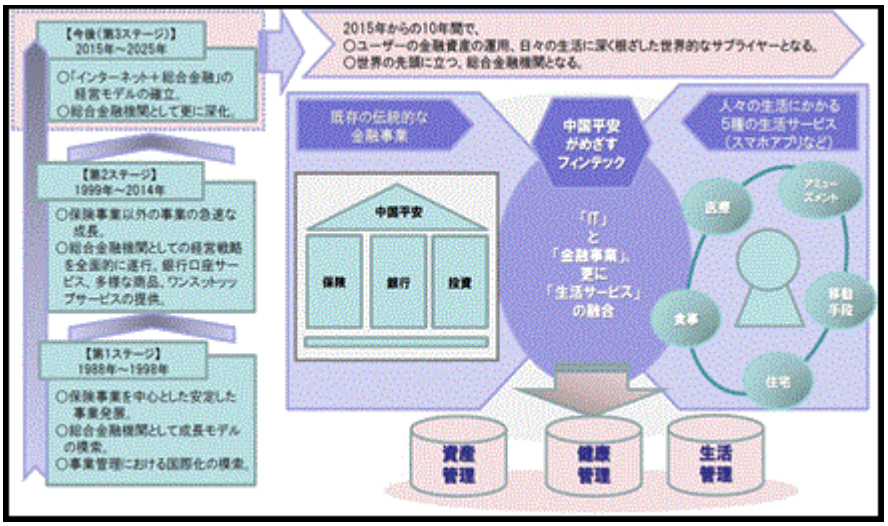
（３）展望

平安保険グループはフィンテック分野に多額の資金を投じ、生体認証（声紋、顔）や AI による予測技術、意思決定技術の更なる進化を進めている。また、インフルエンザな

[2] ビジネス+IT「大阪ガスのビジネスアナリシスセンター所長が語る、データ分析で犯していた３つの勘違い」https://www.sbbit.jp/article/cont1/26465

どの流行予測のサポート事業を開始するなど、平安保険グループが保有するビッグデータや技術の蓄積は、公的保険制度にも広く影響を及ぼしつつある。

図Ⅱ－２：平安保険グループの取り組み[3]




## Ⅱ－４　リテール分野でのビッグデータの顧客志向のための活用事例(Starbucks Rewards)

スターバックスでは、2017 年より同社のプリペイドカード「スターバックスカード」を登録して利用するポイントプログラム「Starbucks Rewards」のサービスを開始し、ビッグデータを活用した「顧客志向」を目指している。

（１）特徴

「Starbucks Rewards」は無料で登録可能なロイヤリティプログラムであり、登録には個人の属性情報が必要となる。「Starbucks Rewards」のデータに基づいて、個人への商品の提案や店舗ごとの戦略などをパーソナライズし、より効果的なマーケティングが可能となるよう利活用している。

（２）実績

スターバックスでは実店舗の顧客体験を最も重要視している。「Starbucks Rewards」の購買データを活用し、個人の好みに基づいた商品開発に取り組んでいる。また、店舗の属する地域への感謝の気持ちを込め、各店舗が自主的に地域向けのキッズパーティーやコーヒーのテイスティングパーティーを開催し、徹底した「顧客志向」に基づくマーケティング戦略を打ち出している。

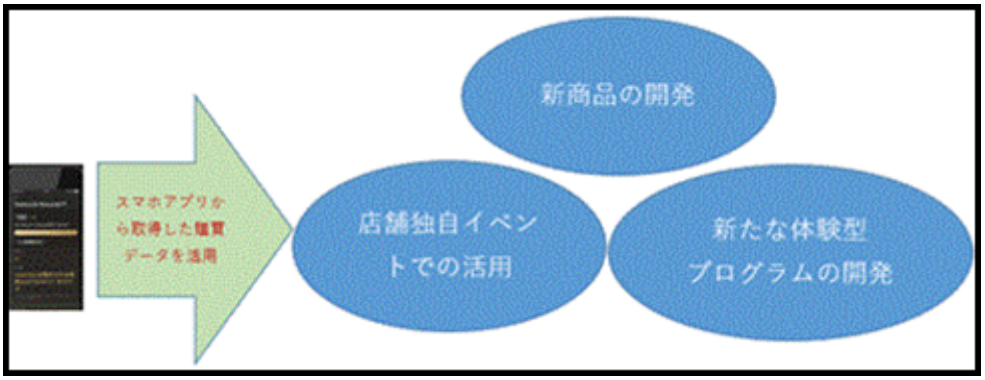
（３）展望

スターバックスでは更なる顧客体験の価値向上のための新たな施策として「ウーバー

[3] ニッセイ基礎研究所「中国 Fintech、平安保険の野望－中国保険市場の最新動向（21)」
https://www.nli-research.co.jp/report/detail/id=53625&pno=2?site=nli

イーツ（Uber Eats)」によるデリバリーサービスを開始するほか、「モバイル・オーダー・アンド・ペイ」をテスト導入するなど、デジタルイノベーションによる新たな顧客体験を目指している。また、スターバックスには 400 万人以上のツイッターのフォロワーが存在する。スターバックスでは常に実店舗での顧客体験を最重要視しているが、多数のフォロワーに向けた SNS での広告戦略も併用することで、「広告を打たなくても集客できる仕組みづくり」を目指している。

図Ⅱ－３：「Starbucks Rewards」の活用イメージ




## Ⅱ－５　国内保険業界におけるビッグデータ活用のための考察すべきポイント

このように他業界・海外保険業界ではビッグデータを活用し、ビジネス変革を実現している例が見受けられる。第Ⅰ章でも述べたように、国内保険会社はビッグデータの活用を進めて行きたいと考えており、先進的な取り組みを推進している海外の保険会社では既に外部のデータを活用し実績を作り始めている状況を踏まえると、国内保険業界でも今後ビッグデータの活用が拡大していくことは想像に難くない。

一方で、国内の保険会社が利用しているデータはまだ限定的であるが、今後デジタル化が進んでいくなか、保険会社にはこれまでとは異なる新たな顧客サービスの提供が求められてくる。保険会社はそのようなデジタル化時代に備え、以下の３つの課題に対応していく必要がある。

（１）社外データから必要なデータを収集する“データ収集”スキーム
（２）分析結果からビジネスニーズを作り上げる“活用ノウハウの蓄積”
（３）ビッグデータ分析に必要な“人材育成”

そこで本論文ではこの３つの課題を、国内保険会社がビッグデータ活用でビジネス変革を起こすために解決すべき重要な課題と定義し、次章以降で現状の整理や活用に向けた具体的な提案を行う。

# 第Ⅲ章　広くデータ収集できるスキームの構築

## Ⅲ－１　データ収集における問題

はじめに、保険会社の保有データの特徴を考える。保険会社の保有データは、保険事務上で必要なデータに限定されており、既契約者情報以外のデータはほとんど保有していない。また、データは契約時断面の状態のまま保持される。一部データは、保険金や給付金の支払、異動発生時等に更新されるが、更新頻度が高いとは言い難い。つまり、保険会社の保有データの特徴として「データのバラエティが少ない」「データの更新頻度が低い」という２点が挙げられる。

こうした特徴は、ビッグデータの活用を限定的なものにしていると考えられる。データ量が少なければ、ビッグデータ解析の幅を狭めることになる。反対にデータのバラエティが多いほど、データ解析により発見される相関関係も多様になり、ビッグデータ活用以前には発見されなかった事象が見いだされるだろう。また、データの更新頻度が高まれば、保険金や給付金の支払いが発生した際に、経緯データを解析することで、被保険者の状態と支払事由の間の新たな因果関係を発見できるかもしれない。

データのバラエティや更新頻度を増やす方法として、自社で収集する方法と、社外データを利用する方法が考えられる。それぞれの効果について、下表にまとめる。

表Ⅲ－１：データ収集の方法と効果

| | 自社で収集 | 社外データの利用 |
|---|---|---|
| バラエティ | △ | ◎ |
| 更新頻度 | ○ | ◎ |

自社で収集する場合、保険契約に関わるデータが主となるため、バラエティを多くすることが難しく、また更新頻度を増やすためには、システム開発等の対応に時間や費用が必要となる。一方、社外データを利用する場合、他業界からデータを収集できるためバラエティは必然的に多くなり、ライフログなど更新頻度が高いデータを活用することも可能である。このことから、「社外データの利用」のほうが、効果が大きいと考えられる。

しかし、社外データを利用する場合、自社の保有情報と外部データを紐付けるキー情報が存在せず、データの紐付けが困難である。また、個人情報保護の観点から、外部データの提供元の審査や、セキュリティ確保といったハードルもある。そのため、外部データを効率的に収集するためには、データを一元的に収集、管理、提供するようなサービスの利用が有効である。

近年では、機械学習やAIの発達により、大量なデータの分析・活用が広がっていることもあり、新たな情報流通の仕組みの検討や運用が始まっている。一例として、情報銀行という仕組みの検討・実験が、官民合同で進められており、2019年には三菱UFJ信託銀行が

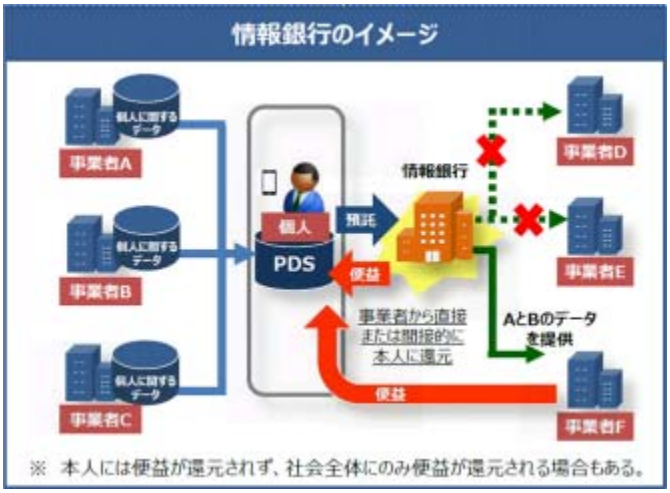
「個人データ銀行」を始める方針を固めている[4]。

## Ⅲ－２　情報銀行とは

情報銀行とは、本人と情報信託機能事業者とのデータ活用に関する契約等に基づき、情報信託機能事業者が本人のデータを管理するとともに、本人の指示またはあらかじめ指定した条件に基づき、本人に代わり、データの性質や利用目的等に照らし、情報提供の妥当性を判断の上、データを第三者に提供するビジネスモデルを指す。本人は、情報を第三者に提供することによって、便益（金銭対価や、サービス等のインセンティブの還元）を受けることができる。

情報銀行は、PDS（Personal Data Store）という仕組みを採用している。これは個人が自分の属性、嗜好、行動履歴等のデータを自ら管理し、その情報を企業等に提供できる仕組みであり、情報を提供する個人にとってもメリットがある。PDS を使って情報銀行や企業に情報を提供するだけでなく、反対に個人が企業へデータの開示を要求して、自らのパーソナルデータを取得することもできる[5]。

図Ⅲ－１：情報銀行活用のイメージ




情報銀行において登場する主なステークホルダーの役割は以下の通りである。

・個人

個人情報の発生源であり、その情報の流通可否について、情報銀行に対して、同意や指示を行う主体。

・情報提供事業者

個人情報は、発生源は本人であるものの、データを生成・保管している主体は、法人（企業等）である場合が多い。具体的には、購買データ、検索情報、健康・医療情報等

[4] 日本経済新聞電子版「三菱 UFJ 信託銀が「個人データ銀行」企業に仲介　【イブニングスクープ】19 年にも」
https://www.nikkei.com/article/DGXMZO33063030X10C18A7MM8000/

[5] データ流通環境整備検討会「AI、IoT 時代におけるデータ活用ワーキンググループ　中間とりまとめ（案）」
https://www.kantei.go.jp/jp/singi/it2/senmon_bunka/data_ryutsuseibi/detakatsuyo_wg_dai9/siryou1.pdf

をはじめとして、データそのものは本人以外の主体が保有していることが一般的である。そこで、情報銀行がデータを流通させるにあたっては、本人の同意のみならず、情報提供元の同意およびデータ提供に係る協力が必要となる。

・情報銀行

情報銀行としてサービスを提供する主体。本人および情報提供元からの同意取得、データの管理、データセットの作成、データ提供可能先の判断、データの流通、インセンティブの還元等を担う。

・情報利活用事業者

情報信託機能事業者を通じて個人データを取得し、活用する主体。場合によっては、この情報信託機能事業者から、直接、情報提供者である本人にサービス等のインセンティブの還元を行うことも想定される[6]。

### Ⅲ－３　保険会社が情報銀行を利用するメリット

情報銀行を利用することにより、各保険会社が保有するデータのほかに、他業種の様々なデータを取得することが可能となる。これにより今までの観点とは異なるデータ分析に基づいた新しい保険商品の考え方が生まれたり、他業種が持つライフログなどのリアルタイムデータの分析により、生活における様々な因果関係も見えてくると考えられる。また、これまで関係が見いだせなかった業界間との関連から、保険業務にとらわれない他業種とのコラボレーションによる新しい価値を提供することも可能となる。

### Ⅲ－４　生命保険業界におけるビッグデータ活用の展望

上述の通り情報銀行の登場により多種多様なデータが分析できるようになり、それに伴い様々なビッグデータ活用が可能となる。ここで生命保険業界におけるビッグデータ活用について我々が考える今後の展望について述べる。

情報銀行などの中央管理的な組織ができることで、保険会社は今まで取得できなかった以下のような他業界の多様なデータを取得することができる。

表Ⅲ－２：取得可能となるビッグデータ例

| 対象データ | データ内容 | 取得元業種 |
|---|---|---|
| 医療データ | 健康診断データ、ゲノムデータ | 医療機関、健康保険組合 |
| ライフログ | 運動データ、身体データ | 健康保険組合 |
| 金融取引ログ | 口座情報、金融商品取引履歴 | 銀行、証券会社 |
| 購買情報データ | 会員情報、購入履歴 | 会員サービス企業 |

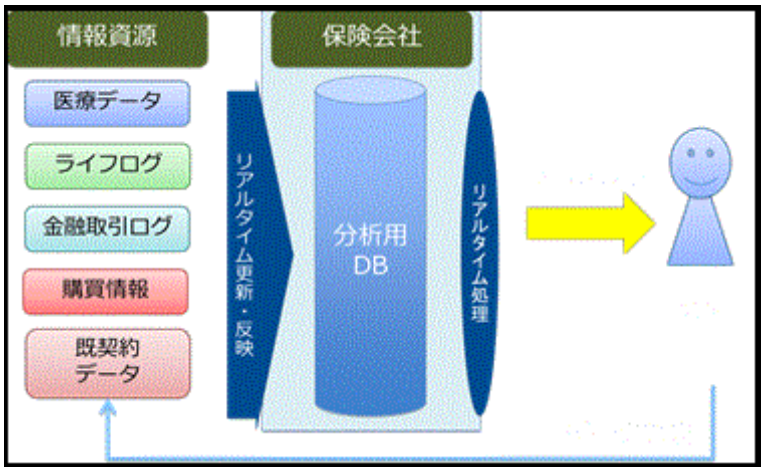
上記のデータを生命保険会社が分析することによって、顧客一人一人に合わせた保険商品のレコメンドなども可能になると考える。例えば、顧客の健康状態を見て、顧客に合わ

[6] The Finance「情報銀行とは～情報銀行ビジネスの動向と今後の展望」
https://thefinance.jp/strategy/181226

せた保険商品を推奨し、その保険商品を顧客にフィットした保険料で提供する。また、金融取引情報や購買情報から顧客の趣向を分析し、その結果に応じて例えば年金保険・学資保険などの保険商品を推奨することができる。

加えて、加入後のデータも収集、分析することで、実施したマーケティングに対する評価を行い、ビッグデータを利用した新しいサービスを顧客に提供できる。

図Ⅲ－２：ビッグデータ活用の展望例




## Ⅲ－５　ビッグデータ活用へ向けた課題

情報銀行によって、ビッグデータの活用は広がることが予想されるが、一方で実現に向けた課題もある。具体的には次の３点であり、生命保険会社はこれらの課題に取り組む必要がある。

（１）データ提供の自発を促すスキームとインフラ整備

情報銀行などに個人のデータを集めるためには、個人の利益や利便性など自発的にデータ提供したくなるようなインセンティブが必須である。そのためのスキーム構築とインフラ整備が今後必要になってくる。具体的にはスマートフォンなどから自動的に情報が連携される仕組みや契約者にメリットを与える仕組みが考えられる。

（２）データ分析チームの組成

収集できるデータの量やバラエティが増加することにより、今まで以上にビッグデータ分析を組織的に取り組む必要がある。収集したデータを継続的に分析し、事業へ活用できるデータ分析チームやデータサイエンティストが必要になる。

（３）収集した様々なデータを活用する最適なシステム

更新頻度の高いデータが増え、そのデータをもとに状況に応じたサービスをリアルタイムで提供することが可能になる。従来のバッチ処理中心のシステム形態ではなく、リアルタイムに処理できるオンライン中心のシステムが求められる。

# 第Ⅳ章　データ分析の実践

本章では、実際に解析の手法の整理、環境構築と実験を行うことを通して、データ分析を実践するにはどのような点がハードルとなるのかを考察する。なお実際に行った手法の詳細な手順、環境、コードについてはすべて Appendix に記載する。

## Ⅳ－１　データ分析の目的

前述のアンケートではビッグデータ活用の検討割合が高いが、実際に活用できている割合は低いというギャップが存在した。データ分析は比較的新しい技術であるため、企業においてビッグデータ活用が進まない理由として、組織面等の他に技術面の問題も考えられる。技術面のハードルについて、実際に解析の手法の整理や環境構築、実験を行い、この結果を通して、データ分析を実践するにはどのような点がハードルとなるのかを考察する。

ビッグデータの特性としてダグ・レイニーが定義した 3V モデルが挙げられる。3V モデルではビッグデータの特性としてボリューム（Volume、データ量)、速度（Velocity、入出力データの速度)、バラエティ（Variety、データ種とデータ源の範囲）があると定義されたが、2012 年にこの定義は次のように更新されている。「ビッグデータは、高ボリューム、高速度、高バラエティのいずれか（あるいはすべて）の情報資産であり、新しい形の処理を必要とし、意思決定の高度化、見識の発見、プロセスの最適化に寄与する[7]」。

さらにビッグデータの基礎技術となっている部分に目を向けると、その基礎となっているのは AI や機械学習と呼ばれるものである。

このため先に述べた実際の検証にあたっては、題材として前述のビッグデータの基礎技術となる機械学習を選択した。機械学習を題材に用いることで、ビッグデータに求められる多次元パラメータ、リアルタイム性（実機上での実行速度）を複数の手法により比較が可能である。

なお、検証の数値の評価のため、本検証では Kaggle と呼ばれるデータ解析のコンペティションに参加し、我々の検証結果が他者との手法と比較した場合にどのような評価となるかも併せて検証する。

## Ⅳ－２　データ分析の実践方法

初めに Kaggle について述べる。Kaggle とはデータサイエンティストや機械学習エンジニアのコミュニティーであり、企業や政府がスポンサーとなり、データ分析の精度を競うコンペティションが実施されている。最も精度の高い分析モデルには賞金が出されるほか、企業から採用のオファーが届くこともある。挑戦者は単純に順位を競うだけでなく、お互いのコードを共有するなど、データサイエンティストの情報共有の場としても活用されている。

[7] https://www.gartner.com/doc/2057415/importance-big-data-definition

当グループでは以下のメンバーにて Kaggle のコンペティションに参加した。

アクチュアリー　　：１名
データアナリスト　：２名
IT エンジニア　　：４名
（全員が生命保険領域）

コンペティションの内容は米国の生命保険会社主催の保険の引受診査を予測するものであり、個人の健康状態のデータから引受リスクを予測する分析モデルを作成する。

分析対象のデータは以下の内容である。

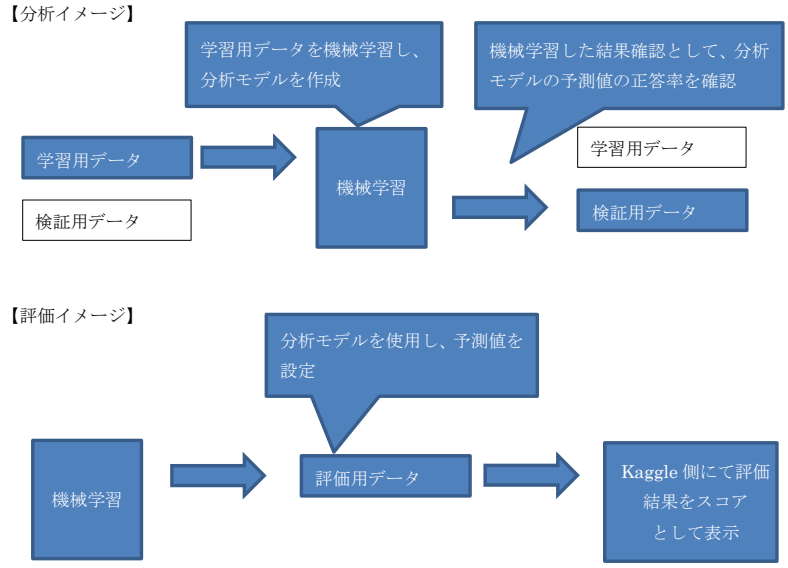
・１行につき１人分の情報があり、全６万件の情報がある。
・１人あたり 100 以上の変数項目があり以下の項目が説明変数、目的変数となっている。
・説明変数：身長、体重、年齢、BMI、病歴などの健康状態に関する項目
・目的変数：リスクを８段階に分けた“Response”という項目

※説明変数とは機械学習で予測するために使用する変数のこと。目的変数とは予測される変数のこと。

手順として、データの 75%（学習用データ）を用いて機械学習を行い、分析モデルを作成する。作成した分析モデルを用いて残り 25%のデータ（分析用データ）に対し“Response”の値を予測する。予測した“Response”値と実際の“Response”値の正答率が分析モデルの精度として算出され、分析モデルの精度を確認する。

その後“Response”値が秘匿されている評価用データ 10,000 件を用いて“Response”値を予測し、Kaggle 側にて正答率を測定した結果をスコアとして表示する。分析モデルはランキング形式でスコア順に並ぶため、コンペティションの参加者はより高スコアとなる分析モデルを作成することが目標となる。

図Ⅳ－１：今回のデータ分析作業の流れ



次に当グループの参加者が機械学習についての知識のない状態からどのような手順でコンペティションに参加していったかを述べる。

まず、知識レベルを統一するために以下の入門書をコンペティションの参加メンバーで各自読み進め、機械学習の手順とアルゴリズムについて理解を深めた。

・「Python で始める機械学習」

～Andreas C. Muller、Sarah Guido　著、中田 秀基　訳～

機械学習のアルゴリズムの種類等を把握したのち、Kaggle の課題を題材に、次の手順で機械学習を実施した。

（1）データの要約（担当者：アクチュアリー、データアナリスト）

データの要約統計量・相関係数を見て、データの特徴をつかむ。欠損値、外れ値や、相関の高い項目を検知し、データ前処理の方針を決定する。

（2）データの前処理（担当者：アクチュアリー、データアナリスト）

以下の４ステップでデータの前処理を実施した。

１．　欠損値を平均値で置換

今回のデータは正規化済みであり、欠損値がないことを確認した。

２．　スケーリング

数値項目を平均:０、分散:１で標準化した。

３．　ダミー変数化

数値以外の項目を０、１で判定するためのダミー変数に変換した。

４．　特徴量エンジニアリング

分析の精度を向上させるため、説明変数に年齢×BMI 等の項目を追加した。

（3）分析モデル構築（担当者：データアナリスト、IT エンジニア）

分析モデルとして使用する以下の種類のアルゴリズムをピックアップした。

・サポートベクターマシン
・ロジスティック回帰
・K 近傍法
・ニューラルネットワーク
・ランダムフォレスト
・勾配ブースティング
・XG ブースティング

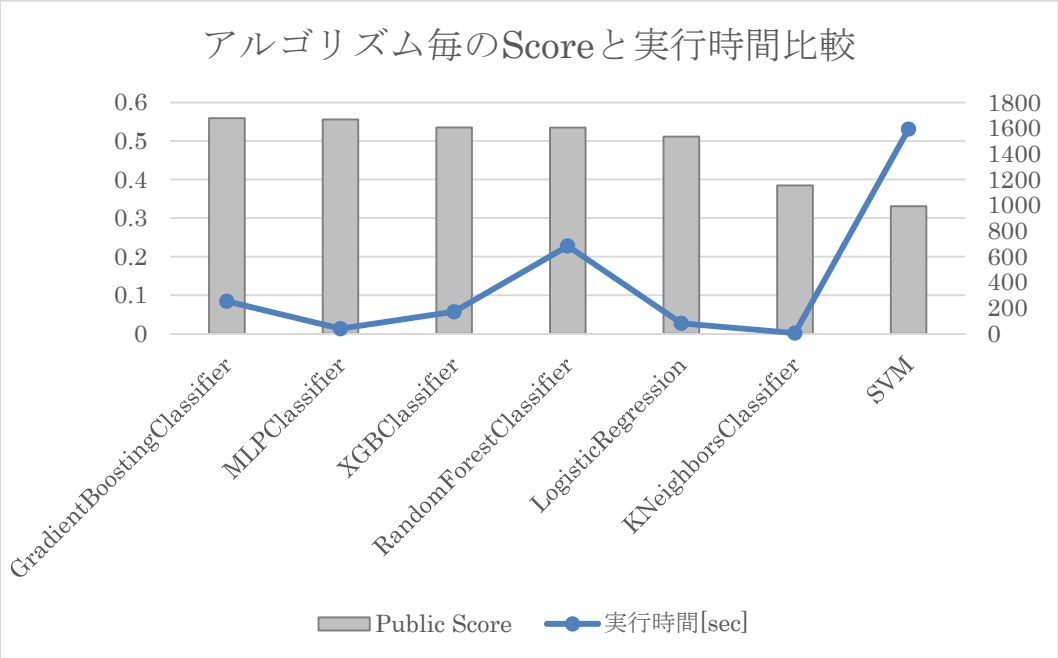
各々のアルゴリズムについて、ハイパーパラメータをグリッドサーチ（パラメータの全組み合わせを試す）で決定し、分析モデルを構築した。

（4）分析モデル評価（担当者：データアナリスト、IT エンジニア）

それぞれの分析モデルで一番精度の高いハイパーパラメータを指定した場合の正答率、実行時間は以下の表の通りとなった。

表Ⅳ－１：アルゴリズムごとのスコアと実行時間

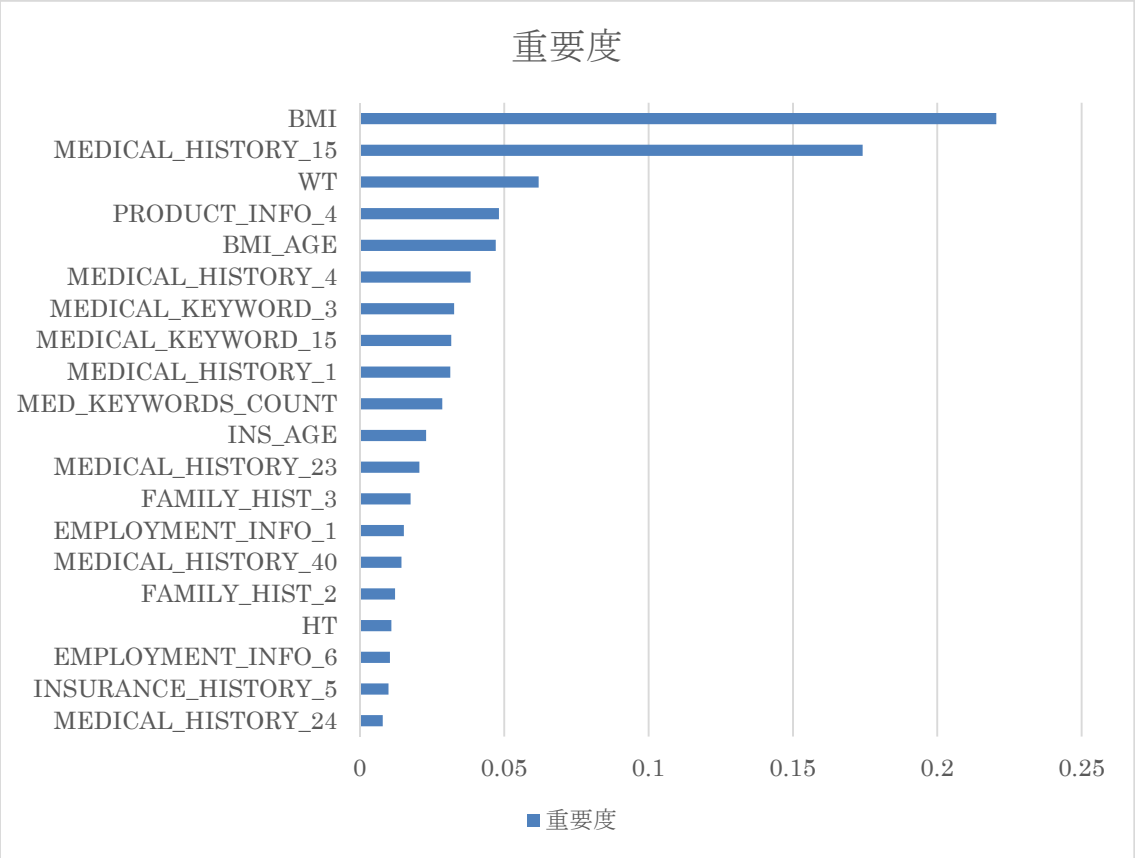
| アルゴリズム | Kaggle の Public Score | 実行時間[秒] |
|---|---|---|
| GradientBoostingClassifier | 0.55911 | 252.68 |
| MLPClassifier | 0.55570 | 38.89 |
| XGBClassifier | 0.53512 | 170.95 |
| RandomForestClassifier | 0.53451 | 682.50 |
| LogisticRegression | 0.51134 | 80.98 |
| KNeighborsClassifier | 0.38492 | 4.65 |
| SVM | 0.33083 | 1592.39 |




また、説明変数の各項目が機械学習においてどの程度重要であったかの結果を得た。

以下の図の通り、今回の分析データにおいては、BMI と Medical_History15 という項目が大きな比重を占めていることが分かった。

図Ⅳ－２：勾配ブースティングモデルにおける説明変数の重要度




## Ⅳ－３　データ分析の実践結果

今回、我々が達成した最も高い Public Score は、勾配ブースティングの 0.55911 である。これは Kaggle の Public Score ランキングにおいて 2,619 チーム中 2,000 位に相当する。

| 順位 | Public Score |
|---|---|
| 1 | 0.68325 |
| 2 | 0.68187 |
| 3 | 0.68135 |
| ・・・ | |
| 1,999 | 0.55930 |
| | **0.55911** |
| 2,000 | 0.55892 |

0.55911 というスコアは、ランダムに分類した場合の正答率 0.125 に比べれば、はるかに高い精度で予測可能なモデルを構築できたことを意味する。

一方、Kaggle の Public Score ランキング内での成績として考えると、下位 25%のランクであり、惨敗と言える。ランク上位の手法について調査してみたところ、複数の機械学習モデルを組み合わせるアンサンブル学習という手法を使用していることが多いが、特に大

きくスコア向上に寄与する手法として、分類モデルではなく回帰モデルを構築し、回帰分析結果として得られた連続値を、目的変数１～８に区切る境界値をチューニングすることで、より精度の高い結果を得るというものがあった。

上記のように、Kaggle で上位を取得するためには、コンペティションで好成績をあげる定石や、目的変数に対する前処理・チューニングを行うといった、高度なノウハウを研究する必要がある。

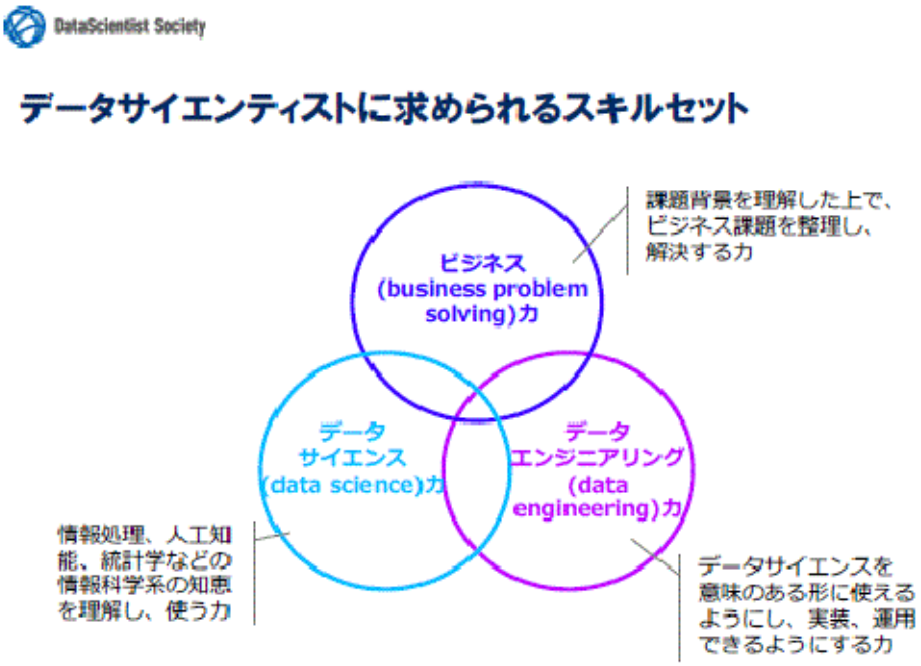
### Ⅳ－４　データ分析を実践することで得られた考察

当節では保険会社としてビッグデータのデータ分析に取り組むに当たり考慮すべき点を、我々の機械学習の実践過程および結果から考察する。

（１）データ分析に必要なスキル要素

図Ⅳ－３に示すようにデータサイエンティストにはビジネス力、データサイエンス力、データエンジニアリング力の三要素が必要であるとされている。

図Ⅳ－３：データサインティストに求められるスキルセット[8]




ビジネス力はドメイン知識、すなわち、保険業界および各社における知識、経験とも言い換えられるであろう。今回の Kaggle の題材は生命保険の引き受け時のリスク区分を８段階で予測する、というものであった。生命保険の知識があれば、これは契約時の査定、すなわち、保険契約を引き受ける、特別条件を付ける、契約を引き受けない、などの分類であろうと理解できる。そして、それには健康状態や年齢が重要な要素になる

[8] 一般社団法人データサイエンティスト協会「データサイエンティストスキルチェックリスト」
https://www.datascientist.or.jp/common/docs/skillcheck.pdf

ということが容易に類推できる。より複雑な問題であればビジネス力は分析の段階でも結果を実際にビジネスに活用する段階でも、より重要なスキルとなる。

次にデータサイエンス力は、統計学に代表される機械学習を支える理論的な背景を理解する力である。機械学習で用いられるアルゴリズムの背景にある数学的理論は簡単ではなく、統計学の基礎知識が不足していると感じるメンバーもいた。データ分析を単純に実行するだけであれば、理論的な裏づけは必ずしも必要ではないが、より複雑な結果を説明し、ビジネス上のアクションに繋げるためには論理的な裏づけも重要になってくるであろう。

データエンジニアリング力は、いわゆるプログラミングスキルに近く、データの前処理で特に重要である。前処理はデータ分析の全工程の８割を占め、必要不可欠な工程であるとも言われる。今回取り組んだ Kaggle の題材では、前処理の必要性は少なかったが、それでも Python に不慣れなメンバーは苦労した。

以上のように、我々が機械学習を実践するなかでも三要素は実際に重要であることが分かった。より良いデータ分析を行うには、これらのスキルセットをより高いレベルで兼ね備えることが必要であるが、一人では限界があるであろう。保険会社においては例えばアクチュアリーと IT エンジニアを中心としたデータ分析チームを構成することも一つの案であろうと考える。アクチュアリーはその実務・教育課程において保険の知識および統計学の知識を身につけている。高度なプログラミングスキルは持っていないかもしれないが、データ分析業務に専業する IT エンジニアが同じチームにいれば、その問題は解消できる。

最後にリーダーシップとコミュニケーションスキルの重要性についても付け加えておきたい。データ分析業務は複雑であるがゆえに、その結果として得られた知見をビジネスに結び付けることが難しい場合が多い。ビジネスマンとしての基礎スキルではあるが、データ分析は専門性が高い業務であるがゆえに見落とさないようにしたい。

（２）予測性能と説明可能性

予測モデルの良し悪しを決めるのは予測の性能（精度）だけでなく、そのモデルの予測方法を説明できるかどうかも重要な要素である。機械学習を用いると人間では到底扱えないほどの多くのプロセスを経て予測モデルを構築することもできる。これは予測の性能を高める一方、その予測プロセスの説明を難しくする。例えば、今回の Kaggle のケースでいうと勾配ブースティングが最も良い性能を示したが、このモデルによる予測結果がなぜそうなるのかを説明するのは簡単ではない。一方で、シンプルなアルゴリズム、例えば k-近傍法によるモデルは性能は低いかもしれないが予測結果を説明するのは比較的容易である。ビッグデータによる機械学習モデルでは、一般には説明の難しい複雑なモデルほど高い性能を示す。逆に説明が容易なシンプルなモデルで十分な性能を求めるのは難しい場合が多い。

どちらの要素を優先するのかは用途によっても異なる。例えば既存顧客のうち購入の

可能性が高い層にのみ新商品のダイレクトメールを送る場合などであれば、その精度が重要であり、説明可能性はあまり求められないであろう。一方で、保険料の計算における将来給付の予測であれば、契約者への公平性の面からも説明可能なモデルであることが非常に重要なことは明らかであろう。

つまり、これらの二つの要素のどちらを、どの程度優先するかを考慮することは機械学習の目標設定において重要なことである。それには、機械学習で用いられる各種アルゴリズムの特徴と統計学的な背景を理解しておく必要がある。また、ビジネス上の目的を正しく理解することも重要である。

（3）機械学習を習得するためのガイドライン

我々は先程挙げた初学者向けの書籍を参考に Kaggle のコンペティションを題材に機械学習を実践した。作業上の問題に対しては、その都度インターネット上の情報を用いて対処した。機械学習を最後まで行い、ある程度の結果は得られたが、Kaggle コンペティション上位のスコアには及ばなかった。Kaggle ではほかの参加者が公開しているコードを参照できるため、成績上位者のコードを確認したところ、より高度な前処理や機械学習を層化するという手法をとっていた。

それらの高度な手法を専門書やインターネット上の情報から学ぶことはできるが、それらは数が多く、また断片的である。機械学習の初級から中級に向かう学習方法、すなわちガイドラインが必要であろう。

（4）PC の性能

機械学習はデータの量やアルゴリズムの複雑性によってはメモリと CPU リソースを大量に消費する。また、機械学習の一連のプロセスは適応する前処理との手法や、その組み合わせおよび学習アルゴリズムを色々変えてみて予測性能を漸進的に向上させていく、という反復的なプロセスである。

反復的であるため、一回当たりの処理にかかる時間はその成果に直結する。人件費に比べて PC の価格が下落しているなか、できるだけ高性能な PC を求めることはデータ分析を行う上で、簡単にできる品質向上策であろう。

## Ⅳ－5　データ分析の実践まとめ

本章ではデータ分析を実際に行うことによって、データ分析において必要な環境を整理することを行った。具体的には、まず Kaggle のコンペティションに参加することを通じていくつかの前処理や手法のあり方を学び実践した。いくつかの手法等を実践した。Kaggle 上での結果はランダムに行うときと比べ精度の高い結果が出たが、専門家には及ばない結果であった。この経験をもとに次の３つのことを導いた。つまり、データ分析には統計力、エンジニア力、ドメイン知識を相互に補うチームを構成すること、機械学習を始めるためにガイドラインを設計したほうが良いこと、そして機械学習を行う環境は、良い環境であるほど良いモデルを作成できることを提案する。

# 第Ｖ章　収集したデータを分析できる組織

## Ｖ－１　ビッグデータ人材の重要性

昨年度の IT 研究論文において、保険会社でビッグデータ活用が進まない要因について、ヒト（人材)、モノ（セキュリティ)、カネ（費用対効果）が挙げられた。特に優先して対策すべき課題はヒト（人材）の確保であると述べられている。

今回行った保険会社へのアンケートにおいては、「貴社は今後データサイエンティストの体制をどのようにしたいと考えていますか」という質問に対して、「社内人材を育成・拡大したい」と回答した会社が５割を占めている。また、体制構築することを検討していない会社は０社となっており、各社体制構築を検討していることがわかる。

一方で、社内育成は検討されているものの、内部で育成方法を確立していると回答した会社は１社もなかった。以上より、各社ともビッグデータ人材を育成中ではあるが、まだその手法は発展途上であり、即戦力として活躍できる人材はもとより将来的に業務を担える候補自体が不足していると考えられる。

我々はビッグデータの活用を推進するためにはまず人材の獲得が重要であると考えた。そこで本章では、より人材確保という観点で深堀りし、昨年度の論文では触れられなかった点である人材獲得の具体的な手段・方法について検討する。

## Ｖ－２　具体的な人材獲得の方法

人材を獲得する方法としては、大きく分けて以下の３通りが考えられる。

・組織内部から登用する
・外部から採用する
・外部のサービスを利用する

以下では、ビッグデータ分析を担える人材という点に着目して、それぞれの具体的な方法やメリット・デメリットについて考察する。

### （１）組織内部での登用・育成

#### a　OJT による育成

一般的な社員教育として行われている On the Job Training（以下 OJT）はデータサイエンティストの育成にも有効である。OJT はトレーナー体制を構築することで社内有識者からノウハウ移管を行い、自社メンバー内で育成を進める。また、外部環境の変化に関わらず、自社内で教育を推進することで、計画的な人材育成が可能となる。しかし、データサイエンティストは教育が可能なレベル（棟梁レベル）の社員は限られており、内部人材だけでは OJT をスタートすることができない。また、スキル移管には時間がかかり、効果が表れるまで時間がかかるというデメリットがある。OJT という方法はデータサイエンティストに限らず一般的な手法であり、人材育成としては主流になりえる手

法である。

b　社内部門からの転用

組織内部の人材活用という視点では、社内部門からの転用も検討できる。データサイエンスに興味のある人材を、社内公募制度で募る企業も少数ではあるが出てきている。社内部門からの転用は、データサイエンティストとしての専門性は低いかもしれないが、当該企業のビジネスを理解しているメンバーを登用できるメリットがある。

しかし、社内には適切な人材が限られていて内部人材の登用が難しいケースも多い。データサイエンスに興味を持つ人材をいかに発掘できるかがポイントとなる。

c　社員全員の底上げ

0JT や社内部門からの登用に限らず、社員全員の底上げも必要となる。データサイエンティストは専門部隊であるが、それを会社全体で活用できなければ効果は半減する。社内研修の中にデータサイエンスに関する教育を含めることで、全社的にデータを活用する機運が高まるようになる。

これらの方法は組織内部の人材を利用するため、外部戦力を活用することに比べ安価な可能性が高く、計画的に人材を確保できるという特徴がある。また、自社ビジネスに精通しているため、適用するデータの特性等を熟知している可能性が高い。

一方、現段階では多くの会社で教育をする側の人材が限られており、教育される側のデータサイエンスに関する知識もほとんどないと考えられる。そのため、育成には時間がかかるとともに、教育する側の人材にかかる負担は大きいものと想定される。長期的な視点で人材育成を図っていくことが重要である。

（2）外部からの採用

外部からの人材を採用することでデータサイエンスのスキル定着を進めることができ、ノウハウがたまっていない企業では有効な方策となる。

a　エージェントを利用した採用

一般的な中途採用と同様に、エージェントを利用してデータサイエンティストを採用することが考えられる。即戦力として期待できる反面、業界全体で人材不足であり、現状では大量採用には向かない。また、採用コストも社内登用に比べると高い。優秀な人材を採用することができれば、効果は高い。

b　コンテストを利用した採用

データサイエンティスト独自の採用方法として、Kaggle のようなデータサイエンスコンペティションの成績優秀者を獲得することも可能である。コンペティションでのスコアや順位により、客観的に能力を評価できるメリットがある。

しかし、一般的に好成績を収めている人材を採用することは難しく、高待遇を用意し

なければ求職者から選ばれない可能性が高い。

c　専門性を有した学生の採用

優秀な学生を擁する学会や研究室をターゲットにして能動的に新卒採用する方法である。また、インターンシップ等を行って学生を募集する方法もあり、これは採用後のミスマッチを防ぐ効果もある。一方、実務経験は浅いため、活躍するまでの時間はかかる。

専門性を有した人材の採用は激化しているが、採用できれば即戦力として期待できる。組織の底上げには、採用した人材から内部人材への知識移転が極めて重要である。

（3）外部サービスの利用

最後に、人材採用するのではなく外部のサービスを利用する方法を紹介する。

a　コンサルティングサービスの利用

データ分析の業務についても、コンサルティングサービスを受けることが可能である。外部サービスを使うことでビッグデータ活用に着手し、具体的に検討を進める足掛かりになる。しかし、コストは高額になることが想定され、社内でデータサイエンスの知見が定着しない可能性も高い。

b　大学との共同研究

データサイエンス関連の大学・研究室と産学共同連携する方法である。企業側からは主に業務で蓄積されたビッグデータを提供し、研究機関側ではそのデータを用いた研究を行う。その結果を企業側へフィードバックし業務へ役立てる手法である。データの蓄積と分析というお互いの強みを活かした手法と言える。大学と共同で研究を行うことで、データサイエンスの知見を企業内へ取り込むことが可能となる。

問題点としては、データをどこまで提供することが可能か、分析自体が目的となりビジネスへの適用が重要視されにくい、などが挙げられる。

c　クラウドソーシングの利用

クラウドソーシングを利用したデータ分析も可能である。クラウドソーシングサイトに登録している不特定多数のデータ分析家に対し、個人情報を含まない研究データを提供し、最適モデルを企業へ返却してもらう方法である。優秀なモデルを返却した分析者に対しては、企業から報酬を支払う。つまり、社内の力を借りることなく、報酬を支払うことでデータ分析が可能となる。

社内でデータサイエンティストを抱える必要がなく、多数の分析者に依頼ができるため、早期に結果を得ることができる。

しかし、内部人材が育成されないという点がデメリットである。さらに、近年では企業側からの報奨金も増加しており、コスト増加の傾向は今後も継続するとみられる。

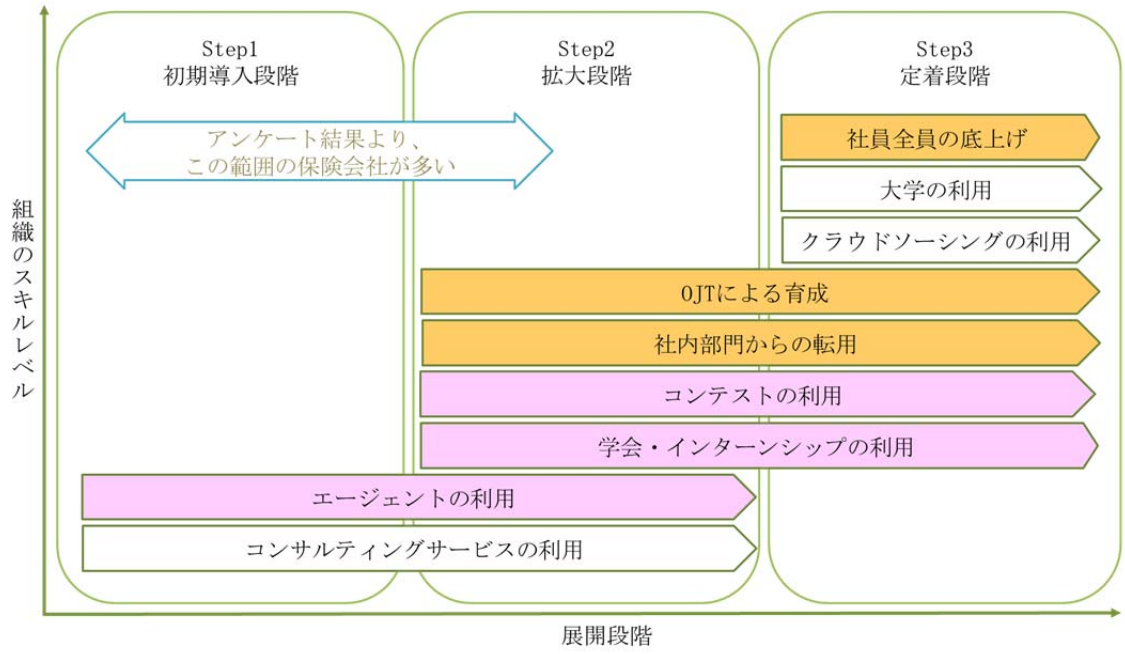
外部サービスを活用する方法は、柔軟な人材活用が可能であり短期間で成果を得ることができるが、コストがかかる、外部依存度が高いと知識が定着しにくいという問題がある。

## Ｖ－３　ビッグデータ活用における人材獲得・育成の提案

円滑に持続的な組織を作り上げていくためには、前節で論じた人材獲得方法を独立に並行して適用していくのではなく、組織の現状に応じて適切な組み合わせを採用していく必要がある。現在の国内保険会社は、ビッグデータ人材の活用という観点ではまだ黎明期と言え、長期的な視点での人材育成が必須である。

そこで我々はビッグデータ人材の獲得・育成ならびに組織が成熟していくためにはどのような過程を辿るのが良いのか検討した。その結果、図Ｖ－１に示すように成長の過程を 3 Step に分けて考えることとし、それぞれの段階で採用すべき手法を整理した。以下その内容を述べる。

図Ｖ－１：ビッグデータ人材の獲得・育成過程




Step1

ビッグデータ人材を登用・育成しようとする初期段階では、自社社員だけ取り組むには難易度が高く限界があると考えられるため、外部の人材をうまく活用していくことが重要である。具体的な手法は以下の通りである。

a　コンサルティングサービスの活用

コンサルティング会社のノウハウを活用して、各社の現況に合わせて、データ分析の取り組み方法を検討する。

b　エージェントを利用した採用

エージェントを通して有識者を中途採用することで、組織化を進める上で必要な人材を確保する。

Step2

拡大段階では、Step1 の取り組みを継続しつつ、長期的視点で社内の人材活用や新卒採用に取り組む。

a　0JT による育成・社内部門からの転用

0JT や社内部門からの転用など社内人材の育成に着手する。

保険会社の場合、アクチュアリーやシステム部門、ビジネス部門と図Ⅳ－３に示すようなデータサイエンティストに必要なスキル（ビジネス力、データサイエンス力、データエンジニアリング力）の三要素を有する人材が元から存在することが多いと考えられるため、この段階で社内人材の活用を推進する。

b　コンテストを利用した採用、専門性を有した学生の採用

長期に安定的に人材確保するために、コンテストや学会、インターンシップを活用した採用も検討すべきである。

Step3

この段階では、データ活用自体を組織文化として定着させ、加えて先端技術を取り込み、他社と差別化を図る。具体的な方法を以下に記載する。

a　大学との共同研究、クラウドソーシングの利用

大学との研究や Kaggle などのコンペティションサイト、クラウドソーシングを活用することで、より分析精度を高める取り組みを行い、競合他社との優位性を高める。

b　社員全員の底上げ

データサイエンティストという専門家を育てるだけでなく、社員全員のデータリテラシーを向上させることで、常にデータに基づく意思決定を行う文化が定着できる。

以上の 3step を経ることにより、データ分析を担う組織の成長・成熟に合わせ適切な人材育成が可能となると考えられる。

# おわりに

従来のデータ分析では、対象となるデータのサンプル数不足によって、新たな傾向を見つけることが困難であった。現在は IoT の普及拡大などにより世の中の情報量は爆発的に増加し続けていることから、対象となるデータのサンプル数を増やし、新たな知見を得ることが可能となってきている。加えて、IT 技術・情報処理技術の進化により、これまでは困難とされていた非構造データですら分析可能となり、いまやビッグデータの活用はデジタル変革を成功させるための重要なファクターとなってきている。我が国の保険業界においてもデジタル化の流れは着実に迫ってきており、まさに“ビッグデータ”が保険業界のあり方を抜本的に変えようとしているのである。

本論文では、国内保険会社がビッグデータ活用でビジネス変革を起こすために解決すべき重要な課題を「広くデータ収集できるスキームの構築」、「データ分析の実践」、「収集したデータを分析できる組織」と定義し、活用に向けた具体的な提案を行った。このデジタル化時代を生き残るために、業界全体でこの課題に取り組んでいくべきではないだろうか。

# 謝辞

当研究の実施に際し、技術支援やヒアリングにご対応いただきました関係各社の皆様、アンケートにご協力いただきましたアクチュアリー会賛助会員各社の皆様、私たちの研究活動を支えてくださった多くの方々に、この場をお借りして深く御礼申し上げます。

また、ご多忙の中、IT 研究大会開催の準備にご尽力いただき、当研究の発表の場を提供いただきましたアクチュアリー会ならびに IT 委員各位に、深く御礼申し上げます。

最後に、IT 研究会への参加を支援いただきました各研究メンバーの所属会社へ厚く感謝いたします。

# Appendix A：　データ分析手順詳細

本章では、Kaggle チャレンジにおける基本方針とした、機械学習による一般的なデータ分析手順を示す。なお、本研究では Python の機械学習ライブラリ scikit-learn と、表形式データ処理用ライブラリ pandas を用いた機械学習を行ったため、各分析手順のサンプルとして、scikit-learn と pandas を用いたサンプルコードを示す。

## Ａ－１　データの要約

データ分析を行う上では、まず分析対象となるデータの内容を確認・分析し、その特徴を掴むことが重要である。特徴として確認すべき主なポイントは、下記５点である。

（１）値種別
（２）要約統計量
（３）欠損値の有無
（４）外れ値の有無
（５）説明変数間の相関係数

以下、各確認ポイントについて概要と具体的な手順を示す。

（１）値種別

データの値は大きく下記３種類に分類できる。

a　ID

データを一意に識別する値である。本質的に目的変数に寄与しないため、学習用データから削除したほうが良い。

b　数値データ（連続値、離散値）

値間に大小関係がある数値である。アルゴリズムによっては変数間の重み付けが均一になるよう、値範囲をスケーリングする必要がある。

c　非数値データ（カテゴリ値）

値間に大小関係が無い数値、または文字列である。特に、文字列データは通常の機械学習アルゴリズムでは処理できないため、事前に数値データに変換する必要がある。数値データへの変換方法としては、各カテゴリに属する・属さないを０／１で表現するダミー変数を生成するやり方が一般的である。図Ａ－１は、「職業」という説明変数を、カテゴリ値「教師」「プログラマー」「公務員」ごとに、新たなダミー変数「職業_教師」「職業_プログラマー」「職業_公務員」を生成する例である。

図Ａ－１　ダミー変数の生成例


| | 職業 |
|---|---|
| 0 | 教師 |
| 1 | プログラマー |
| 2 | 公務員 |

| | 職業_プログラマー | 職業_公務員 | 職業_教師 |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 |

値種別の判別は、データ定義から判断する必要があり、背景となる業界・業務のドメイン知識が要求される。

（２）要約統計量

分析対象データの要約統計量（平均値、標準偏差、最小値、最大値、四分位数）を確認することで、データの特徴を把握する。Python では pandas を使用することで、データの要約統計量を簡単に確認することが可能である。要約統計量の出力は、pandas.DataFrame.describe を使用する。以下に、要約統計量を出力する Python コードと実行結果のサンプルを示す。

```
import pandas as pd
df_raw = pd.read_csv('train.csv')
df_raw.describe()
```

| | Id | Ins_Age | Ht | Wt | BMI | Employment_Info_1 | Employment_Info_2 | Employment_Info_3 | Employment_Info_4 |
|---|---|---|---|---|---|---|---|---|---|
| count | 59381.000000 | 59381.000000 | 59381.000000 | 59381.000000 | 59381.000000 | 59362.000000 | 59381.000000 | 59381.000000 | 52602.000000 |
| mean | 39507.211515 | 0.405567 | 0.707283 | 0.292587 | 0.469462 | 0.077582 | 8.641821 | 1.300904 | 0.006283 |
| std | 22815.883089 | 0.197190 | 0.074239 | 0.089037 | 0.122213 | 0.082347 | 4.227082 | 0.715034 | 0.032816 |
| min | 2.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 | 1.000000 | 0.000000 |
| 25% | 19780.000000 | 0.238806 | 0.654545 | 0.225941 | 0.385517 | 0.035000 | 9.000000 | 1.000000 | 0.000000 |
| 50% | 39487.000000 | 0.402985 | 0.709091 | 0.288703 | 0.451349 | 0.060000 | 9.000000 | 1.000000 | 0.000000 |
| 75% | 59211.000000 | 0.567164 | 0.763636 | 0.345188 | 0.532858 | 0.100000 | 9.000000 | 1.000000 | 0.000000 |
| max | 79146.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 38.000000 | 3.000000 | 1.000000 |

（３）欠損値の有無

欠損値（データ上、ブランクになっている箇所）は機械学習のアルゴリズムでは処理できないため、欠損値を含まないデータに加工する必要がある。まず、欠損値が存在するか否かを確認したうえで、欠損値への対応を行う。

欠損値の有無は、pandas の pandas.DataFrame.isnull と pandas.DataFrame.any を使うことで、簡単に確認可能である。以下に、各説明変数の欠損値有無を出力する Python コードと実行結果のサンプルを示す。以下の例では「True」が出力されている説明変数「Employment_Info_1」に欠損値が存在し、「False」が出力されているほかの説明変数には欠損値が存在しないことを示している。

```
df_raw.isnull().any()

Id                      False
Product_Info_1          False
Product_Info_2          False
Product_Info_3          False
Product_Info_4          False
Product_Info_5          False
Product_Info_6          False
Product_Info_7          False
Ins_Age                 False
Ht                      False
Wt                      False
BMI                     False
Employment_Info_1        True
                        ...
```

欠損値の対応は、後述の通りレコードの削除か、欠損値を何かしらの値で補完する方法が存在する。

（4）外れ値の有無

外れ値の存在は、ほかの正常値の重みを下げてしまい、分析・学習結果に悪影響を与えるため、置換・削除が必要となる。外れ値の存在確認は、分布図・箱ヒゲ図を描くことで確認できるが、実際に外れ値を含むレコードを特定する際は、基準となる四分位数などを前述の要約統計量から取得するのが良い。また、外れ値の対応は、後述に示す通り、欠損値と同様に削除か補完を行うことで対応する。

（5）説明変数間の相関係数

相関関係にある説明変数の存在を知ることは、後述の欠損値補完や次元削減において参考となる。説明変数間の相関係数は、pandas の pandas.DataFrame.corr で取得できる。以下、全説明変数間の相関係数を算出する Python コードと実行結果のサンプルを示す。

```
df_raw.corr()
```

| | Id | Ins_Age | Ht | Wt | BMI | Employment_Info_1 | E |
|---|---|---|---|---|---|---|---|
| Id | 1.000000 | 0.001764 | 0.003674 | 0.005648 | 0.004287 | 0.004870 | |
| Ins_Age | 0.001764 | 1.000000 | 0.008419 | 0.110366 | 0.137076 | 0.096003 | |
| Ht | 0.003674 | 0.008419 | 1.000000 | 0.610425 | 0.123125 | 0.200506 | |
| Wt | 0.005648 | 0.110366 | 0.610425 | 1.000000 | 0.854083 | 0.097917 | |
| BMI | 0.004287 | 0.137076 | 0.123125 | 0.854083 | 1.000000 | -0.005346 | |
| Employment_Info_1 | 0.004870 | 0.096003 | 0.200506 | 0.097917 | -0.005346 | 1.000000 | |
| Employment_Info_2 | 0.003494 | -0.188086 | 0.135446 | 0.068286 | -0.001520 | 0.165485 | |
| Employment_Info_3 | -0.001986 | 0.244600 | -0.139725 | -0.061170 | 0.013766 | -0.219151 | |

## Ａ－２　データの前処理

データ要約において分析対象データの特徴を掴んだうえで、データを機械学習可能かつ効果的な学習を可能とする適切なデータに変換を行う。機械学習に適したデータへの加工作業は一般的に前処理と呼ばれる。

データの前処理では主に下記５つの処理を行う。

（１）欠損値・外れ値の置換／削除

（２）スケーリング

（３）ダミー変数化

（４）特徴量エンジニアリング

（５）次元削減

以下、各前処理作業の概要を示す。

### （１）欠損値・外れ値の置換／削除

欠損値に対する対応には主に「削除」「補完」の２種類があり、それぞれについていくつかの具体的な手法が存在する。

#### a　削除

欠損値を持つレコードを削除してしまうか、または計算上不都合が生じるタイミングだけ無視する方法。実現が簡単である反面、学習データ量の減少や、データ欠損の原因が特定の傾向を持つデータに偏る場合にバイアスを生じさせてしまうデメリットがある。欠損値の代表的な削除方法として「リストワイズ法」が存在する。

①リストワイズ法

単純に欠損値を含むレコードを削除する方法である。pandas の pandas.DataFrame.dropna を使うことで可能である。

```
df_raw.dropna()
```

b　補完

欠損値を何かしらの値で置換する方法である。代表的な方法として「平均値代入法」「回帰代入法」が存在する。

①平均値代入法

欠損値を説明変数の平均値で埋める方法である。pandas の pandas.DataFrame.fillna を使うことで可能である。以下に、欠損値を含む説明変数「Employment_Info_4」を平均値で補完するコードのサンプルを示す。

```
df_raw['Employment_Info_4'] =
df_raw['Employment_Info_4'].fillna(df_raw['Employment_Info_4'].mean())
```

補完が必要な説明変数が大量に存在する場合は、scikit-learn.Imputer を使うことで、一括して補完を行うことが可能である。

```
imp = Imputer(strategy='mean')
imp.fit(df_raw)
imputed_df = pd.DataFrame(imp.transform(df_raw), columns=df_raw.columns)
```

（2）スケーリング

説明変数間での重み付けを均一にするために、数値データの範囲を同じ範囲にそろえる作業をスケーリングと呼ぶ。ランダムフォレストのように、決定木を用いるアルゴリズムでは不要な処理である。スケーリングには「正規化」「標準化」の２つの方法が存在する。

a　正規化

値の範囲を一定の範囲におさめる変換である。実際には[0, 1]か、[-1, 1]の範囲内におさめることが多い。外れ値の影響が大きくなるため、外れ値の除去をあらかじめ行っておく必要がある。

正規化は scikit-learn の preprocessing.minmax_scale を用いることで行うことができる。以下、説明変数「Product_Info_3」を[0, 1]の範囲に正規化するコードのサンプルを示す。

```
from sklearn import preprocessing
df_raw['Product_Info_3'] =
preprocessing.minmax_scale(df_raw['Product_Info_3'].astype(float),
feature_range=(0, 1))
```

b　標準化

値の平均を 0、分散を 1 にする変換である。正規化に比べ、外れ値の影響が小さくなるメリットがある。ロジスティック回帰・SVM、ニューラルネットワークなど、勾配法を用いたアルゴリズムでは標準化を用いたほうが汎化性能が向上する傾向がある。

標準化は scikit-learn の preprocessing.scale を用いることで行うことができる。

以下、説明変数「Product_Info_3」を標準化するコードのサンプルを示す。

```
from sklearn import preprocessing
df_raw['Product_Info_3'] = preprocessing.
scale(df_raw['Product_Info_3'].astype(float))
```

（3）ダミー変数化

ダミー変数の生成は、pandas の get_dummies を使う。get_dummies は、文字列型の説明変数をすべてダミー変数として増幅させ、数値型の変数に対しては何も行わない。以下の例では、2つの説明変数「col1」「col2」を持つデータに対して、ダミー変数化を行う Python コードと実行結果のサンプルである。

```
df = pd.DataFrame({
        'col1': ['a', 'b', 'a'],
        'col2': [1, 2, 3]
    })
df_dummies = pd.get_dummies(df)
```

```
col1_a col1_b col2
0      1      0    1
1      0      1    2
2      1      0    3
```

（4）特徴量エンジニアリング

特徴量エンジニアリングは、説明変数を構築してデータに追加することで機械学習の性能を向上させるテクニックである。主に、カテゴリ変数を数値変数に変換する、1つの説明変数から複数の説明変数に分割する、複数のカテゴリ値を同じ変数に集約する、複数の説明変数を組み合わせて説明変数を作る、などの手法がある。以下、特徴量エンジニアリングの例を示す。

a　カテゴリ変数を数値変数に変換

例：カテゴリ値の出現回数のランキングに変換する（少ない順にランキング）

| 出身国 | → | 出身国_出現回数ランキング |
|---|---|---|
| JP | | 3 |
| US | | 2 |
| UK | | 1 |
| JP | | 3 |
| JP | | 3 |
| US | | 2 |
| JP | | 3 |

b　１つの説明変数から複数の説明変数に分割

例：ブラウザーの UserAgent をデバイス種別、OS、ブラウザーに分割

| UserAgent | → | デバイス種別 | OS | ブラウザー |
|---|---|---|---|---|
| Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0) | | PC | Windows 7 | IE10 |
| Mozilla/5.0 (iPad; CPU OS 11_2_1 like Mac OS X) AppleWebKit/604.4.7 (KHTML, like Gecko) Version/11.0 Mobile/15C153 Safari/604.1 | | タブレット | iOS 11.2.1 | Safari |
| Mozilla/5.0 (Android; Mobile; rv:21.0) Gecko/21.0 Firefox/21.0 Mobile Safari/533.1 | | スマートフォン | Android | FireFox |

c　複数のカテゴリ値を同じ変数に集約

例：日付データを曜日に変換

| 購入日 | → | 購入曜日 |
|---|---|---|
| 2019/01/05 | | Sunday |
| 2019/02/07 | | Thursday |
| 2019/03/11 | | Monday |

d　複数の説明変数を組み合わせて説明変数を作成

例：身長と体重から BMI を作成

| 身長(cm) | 体重(kg) | → | BMI |
|---|---|---|---|
| 167.5 | 63.2 | | 22.5 |
| 183.1 | 68.5 | | 20.4 |
| 160.5 | 71.1 | | 27.6 |

（５）次元削減

説明変数の数が増えると汎化性能向上が難しくなり、学習・分類・回帰の実行時間も悪化する。次元削減は、データの特徴を減らさずに、すなわち学習・分類精度を極力損なうことなく説明変数の種類を減らす方法である。次元削減には、説明変数が絞り込まれることにより、分類モデルを理解しやすくなる（説明能力が向上する）というメリットもある。

次元削減には、大きく特徴選択と特徴抽出の２つの手法が存在する。

a　特徴選択

必要な説明変数のみ選ぶ、または不要な説明変数を除去する手法。単純に説明変数を削減するため、逆に汎化性能が落ちる可能性もある。

b　特徴抽出

複数の説明変数を線形／非線形結合し、１つの説明変数にまとめる手法。前述の特徴量エンジニアリングにおける「複数の説明変数を組み合わせて説明変数を作成」と同じ考え方である。特徴選択と比較して、汎化性能を維持しやすい。

## Ａ－３　モデル構築

モデル構築では、機械学習アルゴリズムをプログラミングで実装し、学習用データを読み込み学習を行い、未知のデータを分類するシステムを構築する。機械学習アルゴリズムの実装のためには、アルゴリズムの数学的な理解と、プログラミングによる実装力の双方が要求され、非常に難易度が高い作業だったが、近年では代表的な機械学習アルゴリズムが既に実装され、ライブラリの形で公開されており、なおかつ学習や学習結果のシステム組み込みが簡易に行えるようになっている。

以下、Python の機械学習ライブラリ scikit-learn におけるモデル構築の初歩的な処理手順をサンプルコードで示す。

```
# pandas, scikit-learn のライブラリをインポート
import pandas as pd
from sklearn.svm import SVC

# 説明変数セットと応答変数セットに分割 (Response が応答変数)
df_train = pd.read_csv('train.csv')
X_train = df_train.drop(["Response"], axis=1) # 説明変数セット
y_train = df_train['Response'] # 応答変数セット

# サポートベクターマシン (SVC) で学習
# ハイパーパラメータは kernel:rbf, C:1, gamma:0.001
clf = SVC(kernel='rbf', C=1, gamma=0.001)

# 学習は fit で行う
clf.fit(X_train, y_train)

# 未知のデータの分類は predict で行う
X_test = pd.read_csv('test.csv')
pred_y = clf.predict(X_test)
```

上記サンプルコードでは、サポートベクターマシンによる機械学習を行っている。サポートベクターマシーンは、scikit-learn では sklearn.svm.SVC として提供されている。そのほか、代表的なアルゴリズムは以下のクラスで提供されている。

| アルゴリズム | クラス名 |
|---|---|
| K-近傍法 | sklearn.neighbors.KNeighborsClassifier |
| ロジスティック回帰 | sklearn.linear_model.LogisticRegression |
| サポートベクターマシン | sklearn.svm.SVC |
| ニューラルネットワーク | sklearn.neural_network.MLPClassifier |
| ランダムフォレスト | sklearn.ensemble.RandomForestClassifier |
| 勾配ブースト法 | sklearn.ensemble.GradientBoostingClassifier |

上記クラスは、すべて fit で学習し、predict で分類するというように使用方法が統一されており、プログラミング時の習得コストが抑えられるようになっている。

各アルゴリズムでは、使用時に前もってハイパーパラメータを設定する必要がある。ハイパーパラメータは分類性能に大きく影響するため、最適なパラメータ選択が重要である。各アルゴリズムにおいて、どのハイパーパラメータが最適であるか、学習・分類対象のデータによって異なるため、各種アルゴリズムとハイパーパラメータの組み合わせを試行する必要がある。

ハイパーパラメータの候補の全組み合わせから最適な組み合わせを探索する手法としてグリッドサーチが存在するが、パラメータの候補値が増えると試行パターンも膨大に膨れ上がるため、非常に煩雑な手法である。scikit-learn では、グリッドサーチを効率よく簡易に実行するための仕組みとして、GridSearchCV クラスを提供している。以下に、GridSearchCV クラスを用いてサポートベクターマシンのハイパーパラメータの最適な組み合わせを探索するコードと実行結果のサンプルを示す。

```
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

# 元データを説明変数セットと応答変数セットに分割
df_raw = pd.read_csv('train.csv')
# 説明変数セット（前処理は省略）
X_train = df_raw.drop(["Id", "Response"], axis=1)
# 応答変数セット
y_train = df_raw['Response']

# SVC のハイパーパラメータの組み合わせを定義
params = [
  {'C': [1, 10, 100, 1000], 'kernel': ['linear']},
  {'C': [1, 10, 100, 1000], 'kernel': ['rbf'], 'gamma': [0.001, 0.0001]}
]

# グリッドサーチによるハイパーパラメータ探索実行
# n_jobs=-1 を設定することで最適な並列実行を行い、実行時間を短縮できる
grid_search = GridSearchCV(estimator=SVC(), param_grid=params, n_jobs=-1)
grid_search.fit(X_train, y_train)

# 最適な組み合わせと、スコアの出力
print(grid_search.best_score_)
print(grid_search.best_params_)
```

```
0.47936645372
{'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
```

## Ａ－４　分析結果評価

学習して構築したモデルの有用性・妥当性を評価するためには、分類性能を見るだけで

はなく、分類のロジック（どの説明変数をどのように分類に使用しているのか）についても確認する必要がある。scikit-learn では、学習して構築したモデルにおける説明変数の重要度（分類結果にどの程度寄与するのかを示す値）を確認する手段が用意されている。

説明変数の重要度は、feature_importances で確認できる。以下、説明変数の重要度をCSV ファイルに出力するサンプルコードと結果を示す。

```
# 学習データでモデル構築
clf.fit(X_train, y_train)

# feature_importances_ は重要度の値のみなので、説明変数名は学習データから取得
df = pd.DataFrame({'Feature' : X_train.columns,
                   'Importance': clf.feature_importances_})
# CSV ファイル出力
df.to_csv('feature_importance.csv')
```

| | Feature | Importance |
|---|---|---|
| 0 | BMI | 0.220457 |
| 1 | BMI_AGE | 0.047085 |
| 2 | EMPLOYMENT_INFO_1 | 0.015254 |
| 3 | EMPLOYMENT_INFO_6 | 0.010408 |
| 4 | FAMILY_HIST_2 | 0.012180 |
| 5 | FAMILY_HIST_3 | 0.017583 |
| 6 | HT | 0.010940 |
| 7 | INSURANCE_HISTORY_5 | 0.009942 |
| 8 | INS_AGE | 0.022962 |
| 9 | MEDICAL_HISTORY_1 | 0.031339 |
| 10 | MEDICAL_HISTORY_15 | 0.174150 |
| 11 | MEDICAL_HISTORY_23 | 0.020626 |
| 12 | MEDICAL_HISTORY_24 | 0.007926 |

# Appendix B： Kaggle チャレンジの実施内容

本章では、Kaggle チャレンジで実施した、機械学習手順を示す。

## B－1 アルゴリズムごとのハイパーパラメータ決定

アルゴリズムごとのハイパーパラメータは、アルゴリズムごとにグリッドサーチを用いて最適なパラメータを決定した。なお、パラメータ探索に使用した学習データは、実行時間短縮のため、Kaggle の学習データ（train.csv）の先頭 10,000 レコードを用い、前処理として欠損値の平均値埋め、文字列値変数のダミー変数化のみ行った。

各アルゴリズムのパラメータ候補と最適なパラメータは下記の通りである。

| アルゴリズム | 候補値 | 最適値 |
|---|---|---|
| GradientBoostingClassifier | n_estimators: [10, 100, 1000]<br>max_depth: [2, 3, 4, 5, 6, 7, 8, 9]<br>random_state: 0 | n_estimators: 100<br>max_depth: 4<br>random_state: 0 |
| MLPClassifier | solver: sgd<br>max_iter: [100, 1000, 10000]<br>random_state: 0 | solver: sgd<br>max_iter: 100<br>random_state: 0 |
| XGBClassifier | learning_rate: [0.01, 0.02, 0.05, 0.1, 0.2, 0.3]<br>max_depth: [3, 4, 5, 6, 7, 8, 9, 10]<br>objective: linear<br>min_child_weight: 360<br>subsample: 0.85<br>colsample_bytree: 0.3 | learning_rate: 0.05<br>max_depth: 6<br>objective: linear<br>min_child_weight: 360<br>subsample: 0.85<br>colsample_bytree: 0.3 |
| RandomForestClassifier | n_estimators: [10, 100, 1000]<br>max_features: [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130]<br>random_state: 0 | n_estimators: 1000<br>max_features: 70<br>random_state: 0 |
| LogisticRegression | C: [0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10]<br>penalty: [l1, l2] | C: 1<br>penalty: l1 |
| KNeighborsClassifier | n_neighbors: [1, 2, 3, 4, 5, 6, 7, 8, 9] | n_neighbors: 9 |
| SVM | kernel: rbf<br>C: 0.00001<br>gamma: 1 | kernel: rbf<br>C: 0.00001<br>gamma: 1 |

### Ｂ－２　前処理内容の決定

データの前処理については、Appendix A にて整理した下記５ステップについて、効果検証を行って決定した。

（１）欠損値・外れ値の置換／削除

（２）スケーリング

（３）ダミー変数化

（４）特徴量エンジニアリング

（５）次元削減

上記の効果有無を検証するため、下記のように前処理の有無の２パターンを全アルゴリズムについて実行し、スコアを比較した。

前処理なし： 学習に必要最低限の前処理のみ（欠損値を平均:０、分散:１で標準化した値に置換、文字列値のみダミー変数化）

前処理あり： すべて実施（欠損値置換、スケーリング、ダミー変数化、特徴量エンジニアリング）

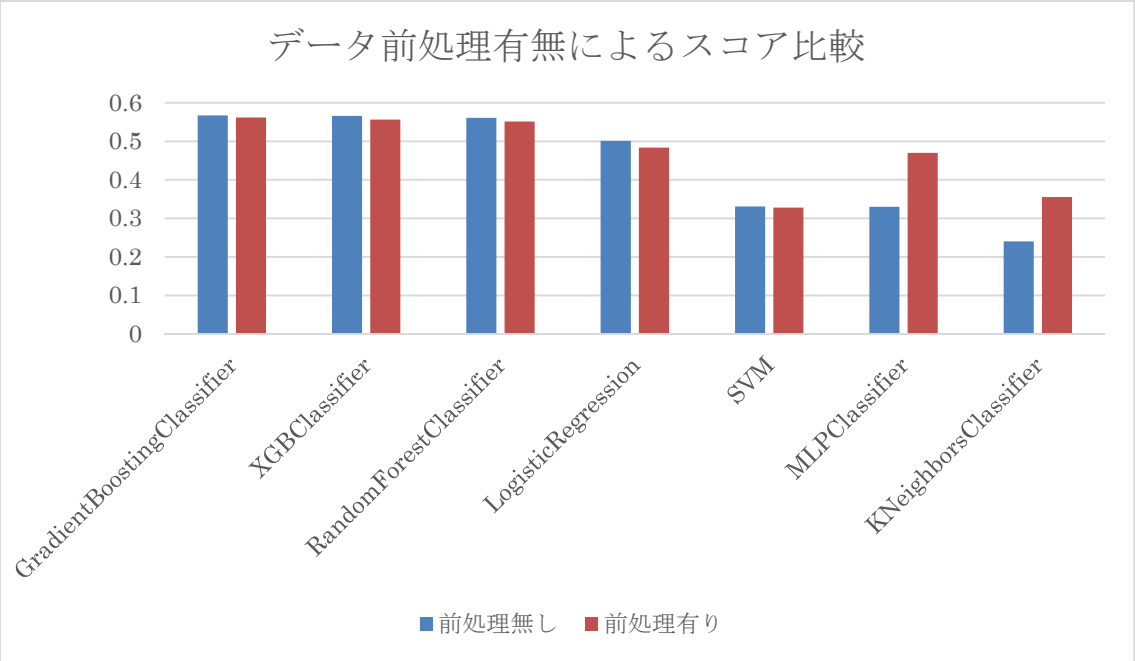
また、スコアの比較にあたっては、下記のように学習用データ・検証用データを作成し、分類結果のスコアを比較した。

学習用データ、検証用データの作成方法：

- Kaggle の学習用データ（train.csv）から先頭 10,000 レコードを抽出し、サブセットを作成
- サブセットの 75%（7,500 レコード）で学習、残り 25%（2,500 レコード）の分類スコアを比較
- 各アルゴリズムのハイパーパラメータは、「Ｂ－１ アルゴリズムごとのハイパーパラメータの決定」で決定した組み合わせを使用

以下に、データ前処理有無によるアルゴリズムごとのスコアの差を示す。

| アルゴリズム | 前処理なし | 前処理あり |
|---|---|---|
| GradientBoostingClassifier | 0.56689 | 0.56149 |
| XGBClassifier | 0.56569 | 0.55639 |
| RandomForestClassifier | 0.56089 | 0.55129 |
| LogisticRegression | 0.50090 | 0.48350 |
| SVM | 0.33083 | 0.32783 |
| MLPClassifier | 0.32993 | 0.46971 |
| KNeighborsClassifier | 0.24025 | 0.35513 |




上記のように、ほとんどのアルゴリズムにおいて、前処理ありによる性能向上が得られなかった。性能が向上しなかった理由として、ダミー変数化による説明変数の過剰な増加が考えられる。説明変数の数を比較すると、前処理なしは 145、前処理ありは 895 と 6 倍以上に増加している。一般的に、説明変数が多くなりすぎると、学習結果のモデルが複雑になりすぎ、分類性能が低下する傾向がある。Kaggle のデータはカテゴリ値の説明変数が 60 項目存在し、ダミー変数化による説明変数の莫大な増加に繋がってしまう。カテゴリ値を離散値に変換することで説明変数の増加を抑止できるが、一方でカテゴリ値に大小の順序関係が生じてしまうため、誤った学習につながる恐れがある。ただし、今回の Kaggle のデータでは、60 項目のカテゴリ値変数のうち、59 項目が離散値として解釈可能な数値データであり、データ前処理なしの実行結果から、誤学習の影響はほとんど無いものと考えられるため、ダミー変数化は必要最低限とした。

そのほかの前処理ステップについては、以下のように実施した

（1）欠損値・外れ値の置換／削除

欠損値を平均0、分散1で標準化

（2）スケーリング

学習用データが正規化済のため実施せず

（3）ダミー変数化

・次項で追加した「Product_Info_2_char」をダミー変数化

・ほかのカテゴリ変数については、ダミー変数化による汎化性能劣化が発生するため、ダミー変数化せず離散値として処理

（4）特徴量エンジニアリング

・年齢×BMIを組み合わせた説明変数追加

・Product_Info_2のカテゴリ値（A1, A2, A3, …, D4, E1）を1桁目（A～E）と2桁目(1～8)の２項目に分割し、それぞれ「Product_Info_2_char」「Product_Info_2_num」の２変数を追加

（5）次元削減

Idを削除

上記のように、Kaggle のデータはほとんどの数値データが正規化されており、かつカテゴリ名も抽象化されているため、前処理による工夫の余地がほとんどなかった。

# Appendix C： 測定環境

本章では、Kaggle チャレンジにおける測定環境および実行時間の計測方法を示す。

## Ｃ－１　測定環境

測定環境には Amazon EC2 を使用しており、Python 環境は Anaconda で構築している。

（１）サーバー

・EC2 インスタンス: c5.xlarge

・vCPU: 4

・Memory: 8GBytes

（２）OS

・Amazon Linux release 2 (Karoo)

（３）Python 環境

・Python 3.6.6

・Anaconda 5.3.1

（４）ライブラリ

・scikit-learn 0.20.1

・scipy 1.1.0

・numpy 1.15.4

・pandas 0.23.4

・py-xgboost 0.80

・memory_profiler 0.54.0

Python の環境構築では、特に各種ライブラリのインストール・管理の観点で、Python 公式インストーラーと pip を用いるか、Anaconda を用いるか、主に２つのインストール方法が存在するが、機械学習用環境を構築する場合は、性能面から Anaconda を用いるほうが良い。機械学習においては scikit-learn や pandas などのライブラリを用いることがデファクトスタンダードとなっており、いずれも数値計算ライブラリである Numpy をベースにして処理を行っているが、Numpy が行列演算用に参照するライブラリである BLAS（Basic Linear Algebra Subprograms）の性能において、pip と Anaconda、それぞれでインストールされる BLAS ライブラリに差異がある。Anaconda でインストールされる BLAS ライブラリは、米 Intel 社が開発した Intel MKL（Math KernelLibrary）という商用ライブラリと同じものであり、pip でインストールされる OpenBLAS に比べて、一般的に高い性能を発揮する。

## Ｃ－２　実行時間の計測方法

実行時間の計測は、アルゴリズムごとに５回実行した上で、最速値を採用した。アルゴリズムの実行時間をｔ、そのうち純粋なアルゴリズムの実行時間をＡ、測定環境上で動作する他プロセス実行による待ち時間をＬとすると、ｔ＝Ａ＋Ｌとなる。このとき、各アルゴリズムにおけるＡは、同一の学習用データ・ハイパーパラメータにおいて一定となり、Ｌはシステムの状態によって変わる。システムの性能評価では、複数のプロセスが並列動作することを考慮するため、Ｌの異常値除去や平均値を求めて評価するのが適切だが、今回の測定ではアルゴリズム自体の性能比較が目的であるため、Ｌが最小となる結果、すなわちｔの最速値を採用するのが適切である。

# Appendix D:　機械学習アルゴリズム

機械学習アルゴリズムは、数多くの手法が研究・提案されている。ここでは、利用実績もある代表的なアルゴリズムをオライリー社の「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」より中心に引用し、いくつか紹介する。

・SVM (Support Vector Machine)
・ロジスティック回帰 (Logistic Regression)
・k 最近傍法 (k-nearest Neighbors)
・ニューラルネットワーク (Neural Network)
・ランダムフォレスト (Random Forest)
・勾配ブースト法 (Gradient Boosting)

| 手法名 | 回帰 | 分類 | 教師 | ハイパーパラメータ例 |
|---|---|---|---|---|
| SVM | ○ | ○ | あり | ・C：誤分類に対するコスト<br>・Gamma |
| ロジスティック回帰 | ○ | ○ | あり | ・C：誤分類に対するコスト<br>・ペナルティ：正則化の方法 |
| k 近傍法 | × | ○ | あり | ・k：データポイントの数<br>・距離測度 |
| ニューラルネットワーク | ○ | ○ | 両方 | ・隠れ層の数<br>・層単位の隠れユニット数 |
| ランダムフォレスト | ○ | ○ | あり | ・決定木の数<br>・深さ<br>・事前の枝刈り |
| 勾配ブースト法 | ○ | ○ | あり | ・決定木の数<br>・学習率<br>・事前枝刈り<br>・深さの最大値<br>・葉ノードの最大値 |

## Ｄ－１　SVM (Support Vector Machine)

（１）アルゴリズム概要

最も一般的な線形クラス分類アルゴリズムの１つ。基本的な仕組みは、データ集合を超平面で２項分類する線形識別器を構築する手法だが、「マージン最大化」という工夫により高い汎化性能を達成し、また「カーネルトリック」という手法により非線形関数による識別器を構築することも可能である。識別性能が優れた機械学習手法の１つとし

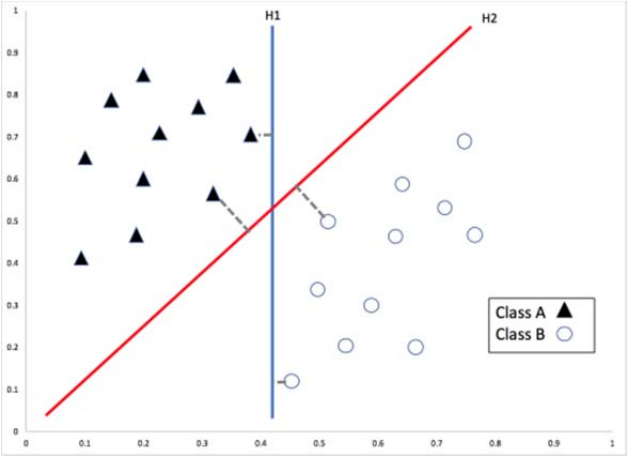
て知られている。

SVM 自体は２項分類器であるため、３種類以上のグループへの分類を行う多項分類問題にはそのまま適用することができないが、複数の分類機を組み合わせる１対１分類法（One-versus-one）や１対多分類法（One-versus-the-rest）というアプローチにより、多項分類問題にも適用可能である。

（２）マージン最大化

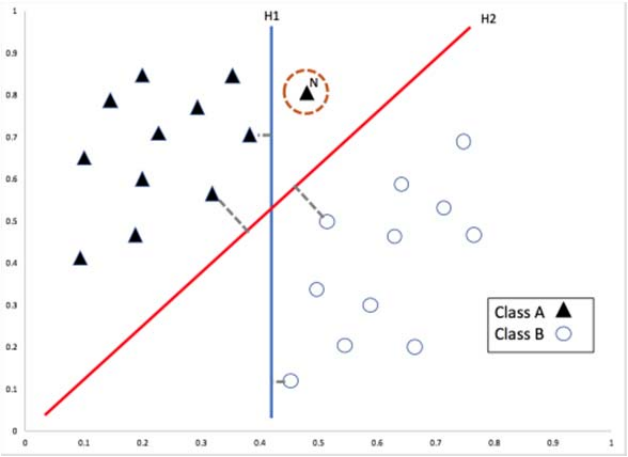
マージン最大化は、２項分類の汎化性能を高めるために採用したアイデアである。基本的な考え方は、分類した２つのクラスに属するデータからの距離（マージン）が最大になるような超平面を構築することである。例えば、図Ｄ－１では直線 H1 と H2 が描かれており、いずれも Class A と B を正しく分類できている。

図Ｄ－１：マージン最大化１




しかし、H1 は H2 よりも Class A,B の点との距離が近い（マージンが小さい）ため、下記の図Ｄ－２のように Class A に属する新しいデータ N を正しく分類できていない。一方、H2 はマージンが大きいため、データ N を正しく分類できている。

図Ｄ－２：マージン最大化２



このように、マージンが大きい超平面のほうが分類の汎化性能が高いと言える。したがって、マージンが最大になるような超平面を算出することが、SVM の基本的な考え方である。２クラスを分類する超平面との距離が近いデータは、全データに対して限られた数のデータである。この一部のデータを「Support Vector」と呼び、SVM の名前の由来となっている。

（３）ソフトマージン

現実のデータにおいては、上記のような超平面により２クラスに線形分離可能なケースは稀である。仮に、２クラスに分類可能な複雑なモデルを構築したとしても、訓練データに過剰に適合し過ぎる（過学習を引き起こす）ことにより、逆に汎化性能を低下させてしまう。SVM では、汎化性能を高めるために誤分類を許容するという工夫を行っており、誤分類を許容するマージンを「ソフトマージン」と呼ぶ。ソフトマージンの算出は、誤分類件数を最小化しつつマージンを最大化する最適化問題であり、下記の数式を最小化する問題として扱われる。

min(1/マージン（サポートベクトルと境界線の距離） + C×誤判別数)

ここで、C は誤分類に対するコストを表すパラメータであり、一般的に C が大きいほど誤分類に厳しくなり、過学習のリスクが高まる。一方で C が小さいほど誤分類に寛容になる。C は極端に大きすぎても小さすぎても汎化性能に悪影響を与えるため、適切な値を選ぶ必要がある。C は SVM の分類性能を決定する重要なパラメータの１つである。

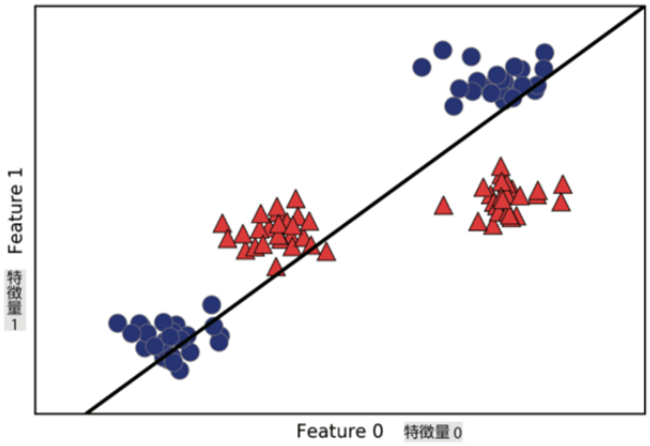
（４）カーネルトリック

カーネルトリックは、SVM を線形分類だけでなく、非線形分類に対応させる手法である。SVM が考案された当初は線形分類にしか対応していなかったが、カーネルトリックが考案されたことで、より複雑なモデルを構築することが可能となり、実データの分類性能が大きく向上した。

カーネルトリックの基本的な仕組みは、元の訓練データが持つ特徴量に対してある種の非線形関数を適用し、変換結果を新たな特徴量として訓練データに追加した上で、線形分離を行うことである。

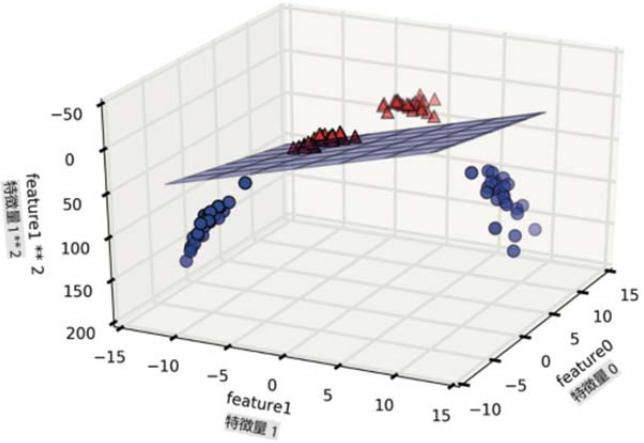
例えば、図Ｄ－３のような特徴量 Feature 0, Feature 1 の２つを持つデータを線形分離する場合を考える。図Ｄ－３のようなデータではどのような直線を引いても、２クラスに分類することは不可能である。

図Ｄ－３：カーネルトリック１




ここで、Feature 1 を２乗した値を新たな特徴量 Feature 1**2 としてデータに追加してみると、図Ｄ－４のように２クラスに分類可能な超平面を構築できる。

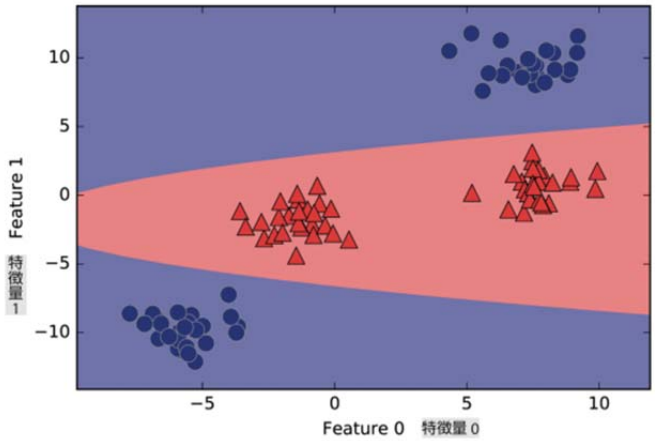
図Ｄ－４：カーネルトリック２




このように、特徴量追加を行うことで非線形分類が可能になったが、その代償として計算量の増加を引き起こす。そこで、特徴量追加による計算を明に行わずに、同様の分類を行う手法が考案された。

上記超平面を元の２つの特徴量に対する関数として描くと、図Ｄ－５のような二次曲線を描くことができる。

図Ｄ－５：カーネルトリック３




このように、元データの特徴量に対する非線形関数の適用結果を追加して線形分類を行うことは、非線形関数を分類曲線として使用することに対応し、特徴量追加による計算を明に行わなくても、より少ない計算量で分類関数を導出することが可能なことを示す。

この元データの特徴量に対して適用する非線形関数を「カーネル」と呼び、実際には特徴量追加を行わずにカーネルの計算のみで分類関数を導出するテクニックを「カーネルトリック」と呼ぶ。SVM のカーネルとしては、可能な限り計算量が少ない関数が望ましく、主に以下の非線形関数が使用される。

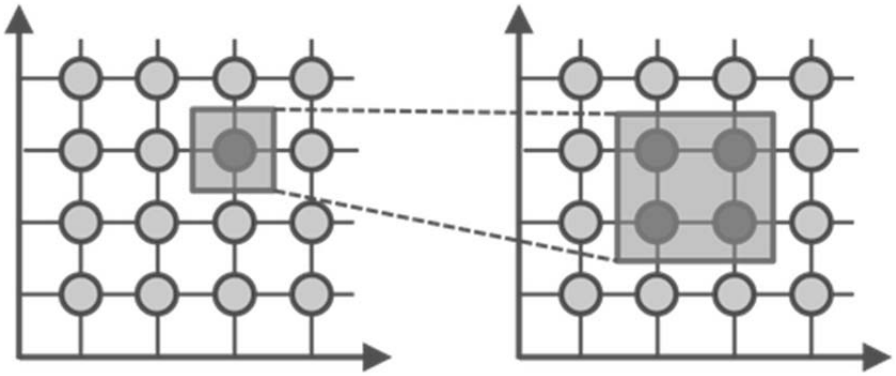
・多項式
・ガウシアン RBF
・シグモイド関数

多くの場合、ガウシアン RBF が最も高い分類性能を発揮することが経験的に知られているが、カーネル固有のパラメータが存在し、適切なパラメータ値を選択することが高い分類性能を発揮するに必要である。

（５）代表的なパラメータと利用指針

正則化パラメータ（C）…SVM ではＣとの最適な組み合わせを決定することが重要となる。一般的にはグリッドサーチにより、C との最適な組み合わせを探索する手法が知られている。

図Ｄ－６：グリッドサーチ



しかし、より少ない試行回数で最適なＣとの組み合わせを決定する手法も報告されており、下記２ステップで行われる。

１．グラム行列の分散が最大となるＣを決定する

２．上記を用いて、SVMの性能が最も良いＣを選択する

## Ｄ－２　ロジスティック回帰（Logistic Regression）

（１）アルゴリズム概要

最も一般的な線形クラス分類アルゴリズムの１つ。多変量解析の一種で、線形回帰分析が量的変数を予測するのに対して、ロジスティック回帰分析は質的確率を予測する。ロジスティック回帰分析は量的変数から質的変数を予測するが、予測する変数の値（１か０かなど）を予測するのではなく、目的変数が１となる確率を予測する。例えば、ある商品の購入有無（「Yes」or「No」）のように、２値しかとりえない値を従属変数の実績値として用い、説明変数を用いてその発生確率を予測する。簡潔に説明すると、それぞれのサンプルが０か１かどちらに属するかの確率を算出して（例え１に属していてもその属する確率は70%なども存在する）、モデルを作成する。

ベルヌーイ分布に従う変数（応答変数が２値の名義尺度）の統計的回帰モデルの一種で、連結関数としてロジット（オッズの対数関数）を使用する一般線形モデルの一種でもある。

ロジスティック回帰を使うことと、見かけ上「回帰」分析に似ている分析方法のため、ロジスティック回帰分析と呼ばれる。

ロジスティック回帰分析は、キャンペーンの反応率や、特定商品の普及率などマーケティングの現場で活用されるほか、防災行政において土砂災害発生危険基準線の確率予測に用いられるなど、気象予測、医療を含む実務分野でも活用されている。

（２）ロジスティック関数（シグモイド関数）

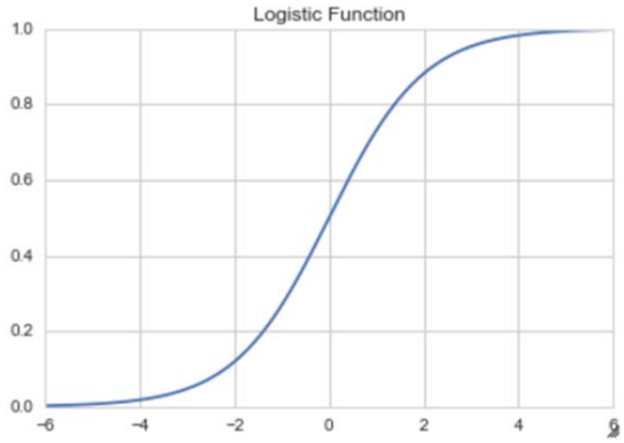
ロジスティック回帰はロジスティック関数もしくはシグモイド関数と呼ばれる以下

の関数（オッズ比の対数関数）の形に目的変数を押し込むことにより、０～１の確率の範囲に変換する。この応答変数が１（確率として）に近いほど、目的変数が１（０か１であるかの１）である可能性が高いサンプルである。

$$\sigma(t) = \frac{1}{1 + e^{-t}} \quad (0 \leqq \sigma(t) \leqq 1)$$

ロジスティック関数の形は図Ｄ－７となる。０～１の値をとり、単調増加の関数である。

図Ｄ－７：ロジスティック関数




（３）フラミンガム研究

1948 年にアメリカ・フラミンガムで始まった疫学研究で、冠状動脈性疾患のリスク因子について調べた研究（フラミンガム研究として有名）でも活用されている。

多くの病気の原因は、複数の因子の組み合わせおよび各因子の程度に依存すると考えられるが、一方で、同程度の因子を持っていても発症する・しない人がいる、といった特徴がある。

フラミンガムの研究では、因子として、年齢・血清コレステロール・収縮期血圧・相対体重・ヘモグロビン・喫煙・心電図所見の７つが検討され、これらの因子への曝露と、発症の割合について、ロジスティック回帰分析によってモデル化された。そして、得られた回帰係数などの情報から、年齢・コレステロール・血圧が高いほど、発症リスクが高くなる、といったことがわかった。

（４）代表的なパラメータと利用指針

正則化の強度（C）…トレードオフパラメータ。小さい C を用いると、データポインタの「大多数」に対して適合しようとする。つまり、正則化（調整）が大きく過学習を防ぐことができる。大きいＣは弱い正則化に対応し、個々のデータポインタを正確にクラス分類することを重視するようになる。

（５）長所と短所

ロジスティック回帰モデルが普及している理由は、あてはめた応答が０と１の間にあり、それゆえ、モデル化するイベントの推定確率が常にたやすく解釈できることがある。

また、クラス分類の問題においては、通常の回帰モデルでは残差が正規分布になることは多くの場合は仮定できないが、その場合でもロジスティック回帰は直接適用できるのもメリットと言える。

## Ｄ－３　k 最近傍法（k-nearest Neighbors）

（１）k 最近傍法（k-nearest Neighbors）

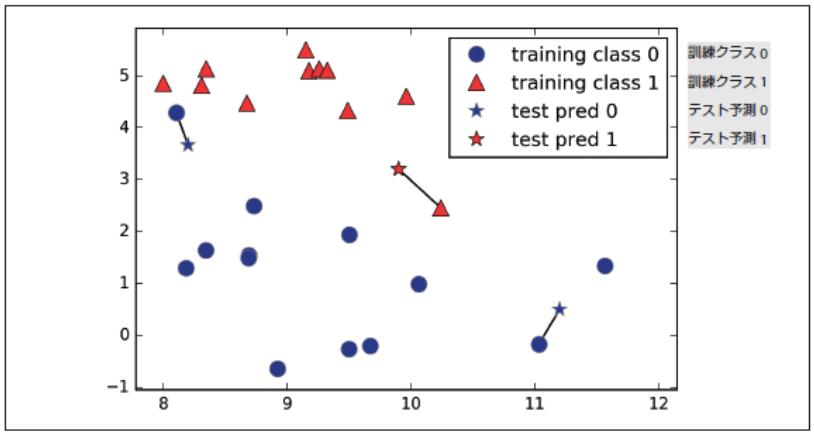
最も単純な機械学習アルゴリズムでクラス分類（２クラス、または多クラス）をするための手法のひとつ。訓練データのうち、最も近い点である k 個の「最近傍点」を検索して、それらの多数決で予測を行う。

一番単純な場合には、１つの近傍点、つまり訓練データに含まれる点の中で予測したいデータポイントに最も近いものだけを見る。予測には、この点に対する出力をそのまま用いる。つまり、k = 1 であれば最も近い１個のデータポイントの目的変数が予測値となる（図Ｄ－８）。

近傍点は１つとは限らず、任意個の、つまり k 個の近傍点を考慮することもできる。これが、k-最近傍法の名前の由来である。１つ以上の近傍点を考慮に入れる場合には、投票でラベルを決める。つまり、個々のテストする点に対して、近傍点のうち、いくつがクラス０に属し、いくつがクラス１に属するのかを教える。そして、最も多く現れたクラスをその点に与える。言い換えれば k-最近傍点の多数派のクラスを採用する。つまり k = 3 であれば最も近い３個のデータポイントを使って多数決が行われた結果の目的変数が予測値となる（図Ｄ－９）。

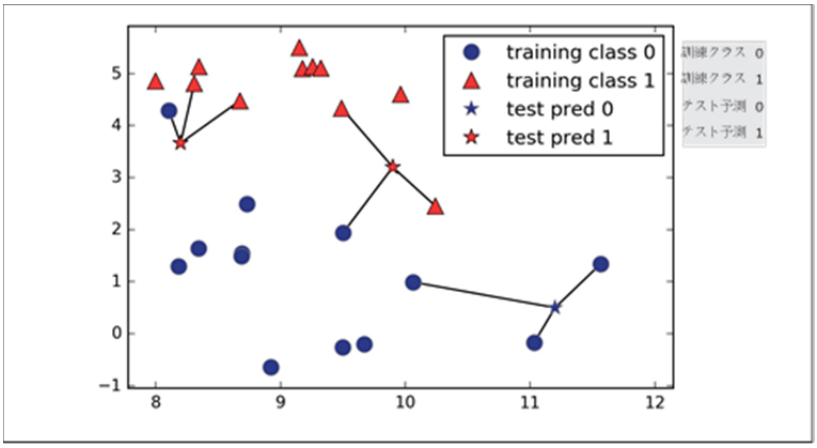
なお、多数決が同数となった場合は、予測対象データとの距離に近いほうが採用される。k = 1（図Ｄ－８）の場合、左上にある★の予測は、最も近い点が class ０のため、class ０と予測される。

図Ｄ－８：k = 1 最近傍点における予測例




k＝3 （図Ｄ－９）の場合、左上にある★の予測は、最も近い３点のうち２点が class １のため、class １と予測される。

図Ｄ－９：k = 3 最近傍点における予測例



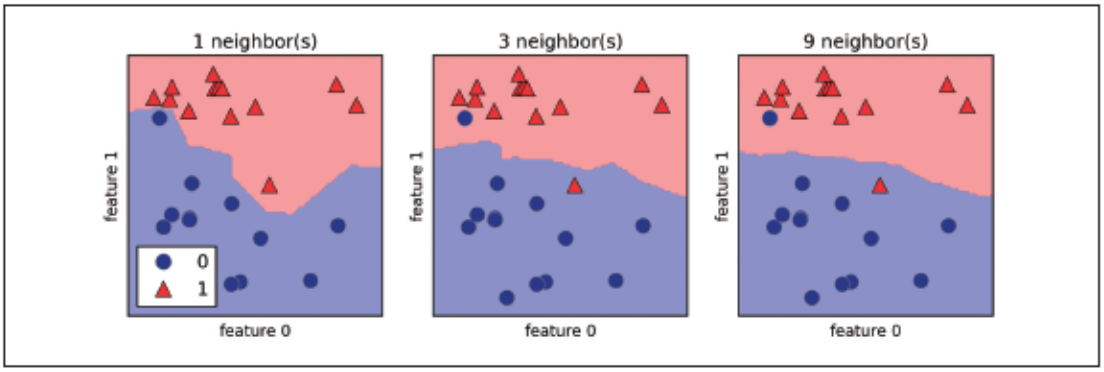

（２）代表的なパラメータと利用指針

近傍点の数（n_neighbors）…予測に使用する近傍点の数。

距離測度…距離測定の方法（ユークリッド距離、マンハッタン距離、など)。デフォルトではユークリッド距離を用いるが、ほとんどの場合はこれでうまくいく。

一般に予測に使用する近傍点の数が小さいと決定境界は訓練データに限りなく近くなり、過学習がおきやすい。近傍点の数を大きくすると制約が大きくなり、滑らかな決定境界となり、シンプルなモデルに近づく。つまり、最近傍点が少ない場合は複雑度の高いモデルに対応し、最近傍点が多い場合は複雑度の低いモデルに対応する。

図Ｄ－１０：異なる最近傍点数に対する決定境界




（３）長所と短所

k-最近傍法の利点の１つはモデルの理解のしやすさにある。また、多くの場合あまり調整しなくても十分に高い性能を示す。より高度な技術の利用を考えてみる前に、このアルゴリズムをベースラインとして試してみるとよい。多くの場合、最近傍法のモデル構築は非常に高速だが、訓練セットが大きくなると（特徴量とサンプルの個数のどちらが大きくなっても）、予測は遅くなる。k-最近傍法アルゴリズムを用いる際には、データの前処理を行うことが重要である。この手法は、多数の特徴量（数百以上）を持つデータセットでは上手く機能しない。また、ほとんどの特徴量が多くの場合０となるような（疎なデータセット（sparse dataset）と呼ぶ）データセットでは特に性能が悪い。

つまり、k-最近傍法は理解しやすいモデルではあるが、処理速度が遅く、多数の特徴量を扱っても予測性能があがりにくいため、実際に使われることは少ない。

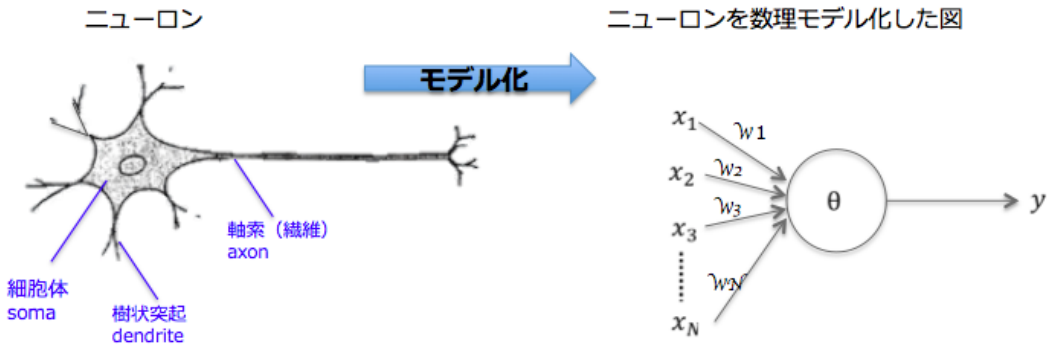
データを理解するために補助的に使うことはありえる。

## Ｄ－４　ニューラルネットワーク（Neural Network）

（１）アルゴリズム概要

ニューラルネットワークとは、人間の脳神経系のニューロン（神経細胞）を数理モデル化したものの組み合わせ（図Ｄ－１１）で、一般的にはディープラーニングと呼ばれるアルゴリズムのこと。人口のニューラルネットワークは生物学的な脳とは異なり、データの伝達方法は事前に層、接続、方向について個別に定義され、それと異なる伝達はできない。

図Ｄ－１１：ニューロンとニューラルネットワークのイメージ図



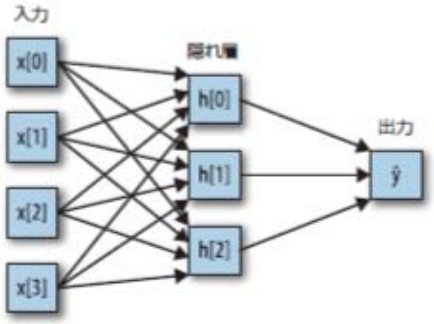

ニューラルネットワークは、教師データ（正解）の入力によって問題に最適化されていく教師あり学習と、教師データを必要としない教師なし学習に分けられる。学習によりネットワークを構成するノード（入力と隠れ層や出力をつなぐ線）の重みを計算（調整）することにより入力と出力の関係性を定義していく（図Ｄ－１２）。学習用の入力データが多ければ多いほど、出力の精度は上がる。ニューラルネットワークにおいて、学習とは、出力層で人間が望む結果（正しい答え、正解）が出るよう、パラメータ（重みとバイアス）を調整する作業を指す。

重み：特定の個体ごとに値を設定する際に使う。重み（重みづけ）はシナプス[9]結合の強さを表す。学習によって重みはシナプスごとにその値が変化する。

バイアス：値を偏らせるために広く同じ値を設定する際に使う。

ニューラルネットワークには、閾値が不可欠。閾値は判定基準で、基本的に変化しないものである。生物学で例えるならニューロンの感度のようなもののこと。

図Ｄ－１２：一層の隠れ層を持つニューラルネットワーク




（２）ニューラルネットワークのパラメータチューニング

パラメータチューニングによるニューラルネットワークの性能向上を多層パーセプトロン（multilayer perceptron：MLP）[10]の手法を引用した図Ｄ－１３と図Ｄ－１４を

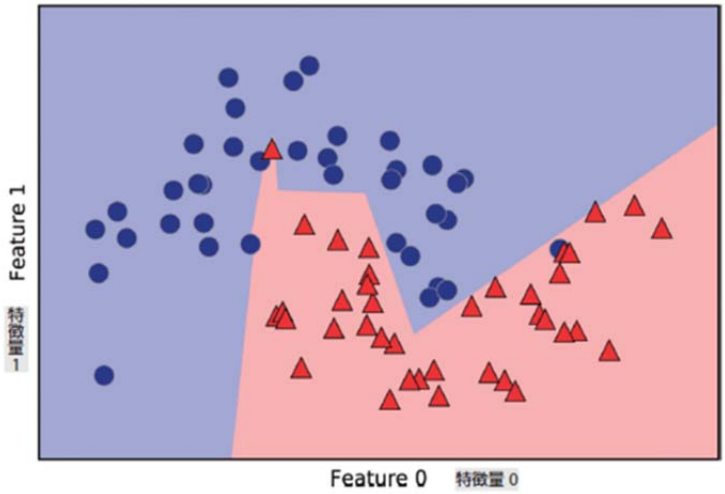
[9] ニューロン同士の結合間には、シナプスが存在する。シナプスは電気信号を化学物質の信号に変換し、情報を伝達する特別な構造になっている。

[10] ユニットの出力をゼロか１かの２パターンとして作成したパーセプトロンといわれるユニットの層を、複数組み合わせることにより作成したニューラルネットワークのこと。

用いて説明する。

例として隠れ層を 10 ユニット使用して２つに分類した結果が図Ｄ－１３である。決定境界が非常にカクカクしていることが見て取れる。

図Ｄ－１３：隠れ層に 10 ユニットを持つニューラルネットワーク




この決定境界を滑らかにする方法は、大きく３つある。

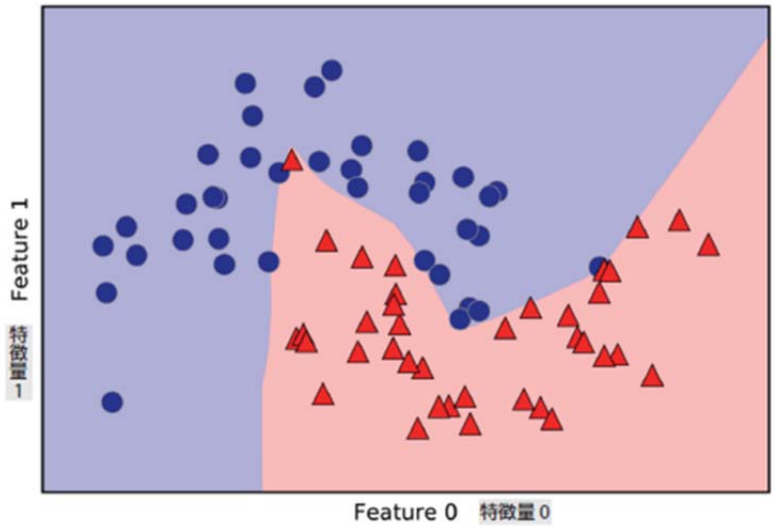
・隠れ層自体を増やす

・ユニット数を増やす

・非線形活性化関数に tanh を用る

図Ｄ－１４において隠れ層を 100 ユニットに増やした例を示す。図Ｄ－１３と比較すれば、その境界がより滑らかになったことが見て取れる。

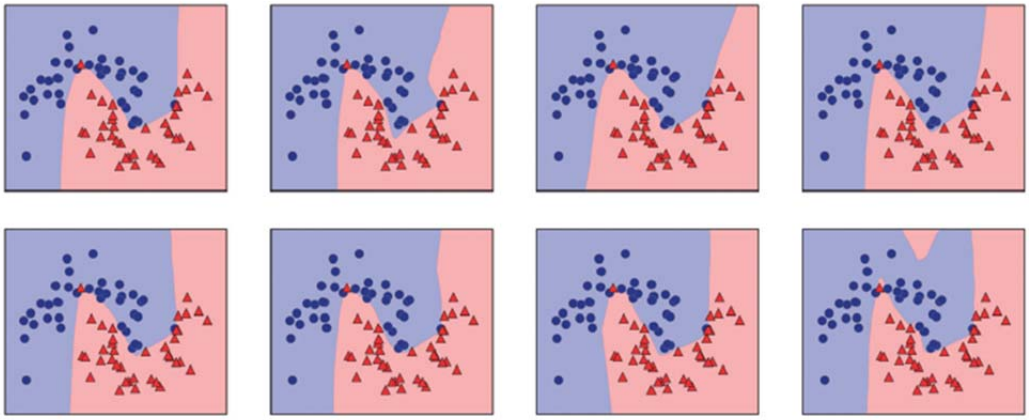
図Ｄ－１４：隠れ層に 100 ユニットを持つニューラルネットワーク




（３）ニューラルネットワークの性質

ニューラルネットワークは学習を開始する前に重みを乱数で割り当てる。この乱数による初期化の影響が、学習されるモデルに影響を与えることが、ニューラルネットワークの重要な性質の１つとしてある。これはまったく同じパラメータを用いても、異なる乱数シードを用いると、図Ｄ－１５のようにまったく異なったモデルが得られることを意味する。ネットワークが大きくなると、複雑さを適切に設定しさえすれば、精度にはそれほど影響を与えないはずだが、（特に小さいネットワークでは）このことを留意すべきだ。

図Ｄ－１５：異なる乱数で初期化された状態から学習されたさまざまな決定境界



（４）代表的なパラメータと利用指針

隠れ層の数と層あたりのニューロンの数（hidden_layer_sizes）…最も重要なパラメータ。隠れ層は１つか２つで始め、あとから拡張していけばよい。隠れ層あたりのノードの数は、入力層と同じくらいにすることが多いが、数千より大きくなることはあまりない。

ニューラルネットワークのパラメータを調整する一般的なやり方は次のようになる。まずは過剰適合できるように大きいネットワークを作って、タスクがそのネットワークで訓練データを学習できることを確認する。次に、ネットワークを小さくするか、alphaを増やして正則化を強化して、汎化性能を向上させる。

（５）長所と短所

ニューラルネットワークの最大の利点は、大量のデータに含まれているデータを費やし、信じられないほど複雑なモデルを構築できることである。十分な計算時間とデータをかけ、慎重にパラメータを調整すれば、ほかの機械学習アルゴリズムに勝てることが多い（クラス分類でも回帰タスクでも）。

これは裏返せば欠点にもなる。ニューラルネットワークは、特に大きくて強力なものは、訓練に時間がかかる。さらに、ここでも見たように、データを慎重に前処理する必要がある。SVM と同様に、データが「同質」な場合、つまりすべての特徴量が同じ意味

を持つ場合に、最も良く機能する。さまざまな種類の特徴量を持つデータに関しては、決定木に基づくモデルのほうが、性能が良いだろう。ニューラルネットワークのパラメータのチューニングは、それ自体が１つの技芸となっている。

## Ｄ－５　ランダムフォレスト（Random Forest）

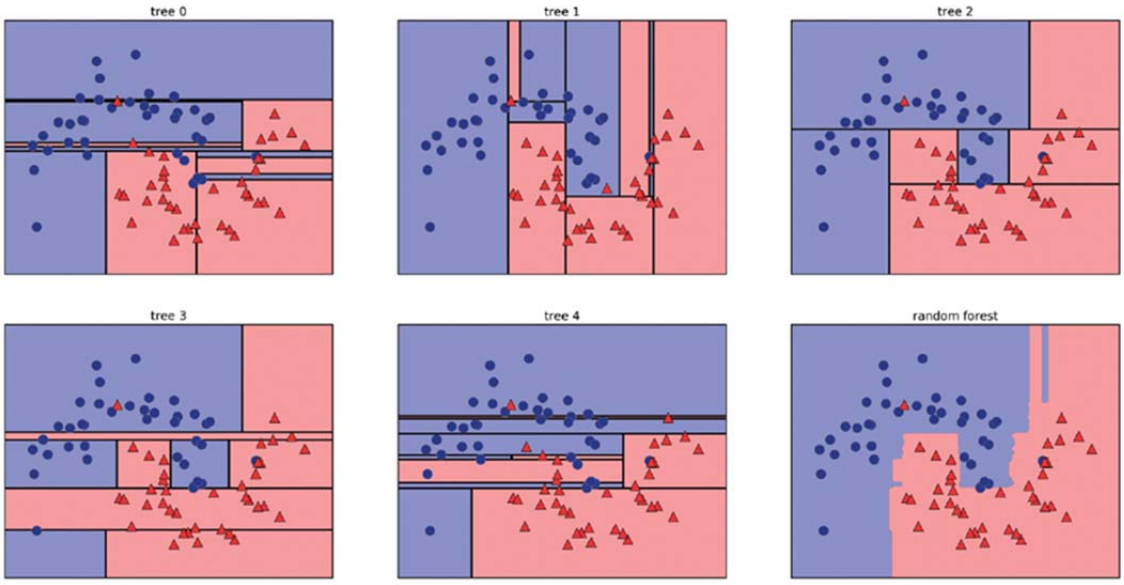
（１）アルゴリズム概要

ランダムフォレストとは、複数の機械学習モデルを組み合わせることで、より強力なモデルを構築する手法であるアンサンブル法の一種で、クラス分類や回帰に関して有効な手法のひとつである。

ランダムフォレストとは、少しずつ異なる決定木をたくさん集めたものである。これは決定木の最大の問題点である訓練データに対して過剰適合してしまうことに対する対応の１つである。ランダムフォレストは、個々の決定木は比較的上手く予測できているが、一部のデータに対して過剰適合してしまっているという考えに基づいている。それぞれ異なった方向に過剰適合した決定木をたくさん作れば、その結果の平均をとることで過剰適合の度合いを減らすことができる。決定木の予測性能を維持したまま、過剰適合が解決できることは厳密な数字で示すことができる。参考として図Ｄ－１６に５つのランダム化された決定木による境界と、それらを平均して得られた決定境界を示す。

この戦略を実装するには、たくさんの決定木を作らなければならない。それぞれの決定木は、ある程度ターゲット値を予測できていて、さらにお互いに違っていなければならない。ランダムフォレストという名前は、個々の決定木が互いに異なるように、決定木の構築過程で乱数を導入していることから付いている。ランダムフォレストに乱数を導入する方法としては、決定木をつくるためのデータポイントを選択する方法と、分岐テストに用いる特徴を選択する方法の２つがある。

図Ｄ－１６：５つのランダム化された決定木による決定境界と、それらを平均して得られた決定境界




（２）ランダムフォレストによるモデルの構築過程

決定木を構築するには、まずデータからブートストラップサンプリング（bootstrap sample）と呼ばれるものを行う。これは、n_samples 個のデータポイントから、交換ありで（つまり、同じサンプルが何度も選ばれる可能性がある）データポイントをランダムに n_samples 回選び出す手法である（復元抽出）。これによって、もとのデータセットと同じ大きさだが、データの一部（だいたい３分の１）が欠け、一部が何度か現れているデータセットが得られる。例えば、リスト['a', 'b', 'c', 'd']からブートストラップサンプリングしてみた結果としては、['b', 'd', 'd', 'c']や['d', 'a', 'd', 'a']が得られる。

次に、この新しいデータセットを用いて決定木を作る。ただし、決定木を作るアルゴリズムを少しだけ変更する。個々のノードで最適なテストを選ぶのではなく、特徴量のサブセットをランダムに選び、その特徴量を使うものの中から最適なテストを選ぶ。特徴量サブセットの大きさは、パラメータ max_features で制御できる。この特徴量のサブセットの選択は、個々のノードで独立に繰り返し行われる。これによって、決定木の個々のノードが異なる特徴量のサブセットを使って決定を行うようになる。

ブートストラップサンプリングによって、ランダムフォレストの中の個々の決定木が少しずつ違うデータセットに対して構築されることになる。さらに、個々のノードでの特徴量の選択によって、それぞれの決定木は異なる特徴量のサブセットに対して分割を行うことになる。これらの機構が組み合わされることで、ランダムフォレスト中の個々の決定木が異なるものになる。

（3）代表的なパラメータと利用指針

構成する決定木の数（n_estimators）…より多くの決定木の平均を取ると、過剰適合が低減されアンサンブルが頑健になるから、大きければ大きいほうがよい。しかし、増やすことによる利益は徐々に減っていくし、メモリの量も訓練にかかる時間も増大する。簡単なルールとしては、「時間とメモリのある限り大きくする」ということになる。

個々の決定木の乱数性（max_features）…小さくなると過剰適合が低減する。一般にはデフォルト値を使うとよい。クラス分類については max_features=sqrt(n_features)、回帰については max_features=n_features となっている。max_features や max_leaf_nodes を追加すると性能が上がることがある。また、訓練や予測にかかる時間を大幅に縮められる場合もある。

max_features を n_features に設定すると、それぞれの分岐でデータセット中のすべての特徴量を見ることになり、特徴量選択時の乱数性はなくなる（ブートストラップサンプリングによる乱数性は残る）。max_features を１にすると、分岐時に使う特徴量選択にはまったく選択肢がないことになり、ランダムに選ばれたある特徴量に対してスレッショホールドを探すだけになる。したがって、max_features を大きくすると、ランダムフォレスト中の決定木が似たようなものになり、最も識別性の高い特徴量を使うので、訓練データに容易に適合できる。max_features を小さくすると、ランダムフォレスト中の決定木は相互に大幅に異なるものとなるが、それぞれの決定木をかなり深く作らないと、データに適合できない。

本質的に、ランダムフォレストは決定木の利点の多くを残したまま、決定木の欠点の一部を補っている。それでも決定木を使う理由があるとしたら、決定プロセスの簡潔な表現がほしい場合ぐらいである。何十、何百もの決定木を詳細に解釈することは不可能だし、ランダムフォレスト中の決定木は、（特徴量のサブセットを使うので）単独の場合よりも深い傾向にある。したがって、もし予測の過程を専門家でない人たちにもわかるように可視化したいのであれば、単独の決定木を使ったほうがよいだろう。

（4）長所と短所

ほとんどの場合、単一の決定木よりも高速、頑健、かつ、精度も高いのが長所である。また多くの場合それほどパラメータチューニングをせずに使えるし、データのスケール変換をする必要もないため、回帰でもクラス分類でも現在最も広く使われている機械学習手法である。

短所としては、ランダムフォレストはテキストデータなどの非常に高次元で疎なデータに対してはうまく機能しない傾向にある。このようなデータに対しては線形モデルのほうが適している。一般にランダムフォレストは、非常に大きいデータセットに対しても機能するし、強力な計算機では複数の CPU を用いて簡単に並列化できる。しかし、ランダムフォレストは線形モデルよりも多くのメモリを消費するし、訓練や予測も遅い。実行時間やメモリが重要なアプリケーションでは、線形モデルを使ったほうがよいだろ

う。

## Ｄ－６　勾配ブースト法（Gradient Boosting）

（１）アルゴリズム概要

ランダムフォレストと同様に、複数の決定木を組み合わせてより強力なモデルを構築するアンサンブル手法。回帰にもクラス分類にも利用できる。

ランダムフォレストと対照的に、勾配ブースティングでは１つ前の決定木の誤りを次の決定木が修正するようにして、決定木を順番に作っていく。正解値と弱学習器の予測値との差を負の勾配とみなして、それを最小化するように逐次的に弱学習器を学習させていく。それぞれの学習器の正解率を元に重み付けを決定し、最終的にたくさんの学習器の重み付き多数決で分類を行う。

学習器デフォルトでは、勾配ブースト法には乱数性はない。その代わりに、強力な事前枝刈りが用いられる。勾配ブースト法では、深さ１から５ぐらいの非常に浅い決定木が用いられる。これによって、モデルの占めるメモリが小さくなり、予測も早くなる。勾配ブースト法のポイントは、浅い決定木のような、簡単なモデル（弱学習器）を多数組み合わせることにある。それぞれの決定木はデータの一部に対してしか良い予測を行えないので、決定木を繰り返し追加していくことで、性能を向上させる。

勾配ブースト法は、機械学習のコンペティションでしばしば優勝しており、産業界でも広く使われている。ランダムフォレストに比べるとパラメータの影響を受けやすいが、パラメータさえ正しく設定されていれば、こちらのほうの性能が良い。

なお著名な手法に XGBoost が存在する。これは Gradient Boosting の派生であり、概要は同様のものである。

（２）勾配ブースト法によるモデルの構築過程

図Ｄ－１７の手順で勾配ブースト法によるモデルを作成する。

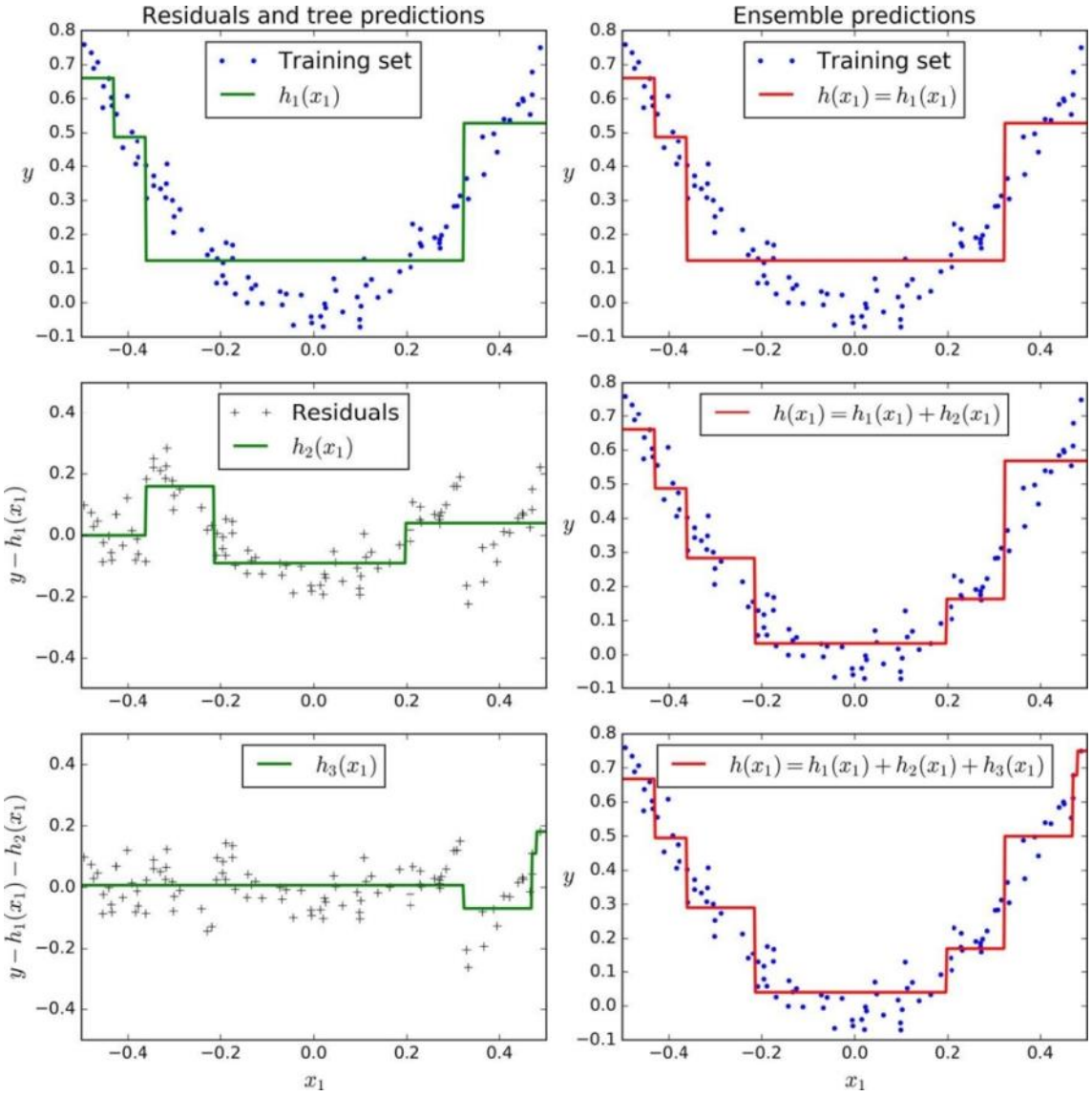
①（浅い）決定木で学習して予測器を作る（図Ｄ－１７　１段目左）。

②残差誤差に対して同じく（浅い）決定木で学習する（図Ｄ－１７　２段目左）。

③図Ｄ－１７の２段目の右図は一回目の決定木と二回目の決定木のアンサンブル（加算）したもの。浅い決定木を２つ組み合わせただけでも、より正確に予測できていることがわかる。

④３段目以降も同様の処理を繰り返すことで、より誤差の少ないモデルが作成される。

図Ｄ－１７：勾配ブースト法によるモデルの構築過程




（３）代表的なパラメータと利用指針

決定木の数（n_estimators）…ランダムフォレストの場合には大きければ大きいほど良かったが、勾配ブースティングの場合には大きくすると、訓練セットに対する過ちを補正する機会が増えるので複雑なモデルを許容することになり、過剰学習を招く。決定木の数は時間とメモリ量で決めておいて、学習率に対して探索を行う方法がよく用いられる。

学習率（決定木の誤りを補正する度合い）（learning_rate）…個々の決定木が、それまでの決定木の過ちをどれくらい強く補正しようとするかを制御するパラメータ。小さくすると同じ複雑さのモデルを作るにはよりたくさんの決定木が必要になるため、決定木の数と強く相関している。

深さの最大値（max_depth）…事前枝刈り（過剰学習を防ぐためのパラメータ）のひ

とつ。個々の決定木の複雑さを減らす。決定木の質問の最大数。一般に勾配ブースティングでは非常に小さく設定され、深さが５以上になることはあまりない。

葉ノードの最大値（max_leaf_nodes）…事前枝刈りのひとつ。個々の決定木の複雑さを減らす。ノードの最大数。

勾配ブーストとランダムフォレストは、同じようなデータを得意とするので、一般には、ランダムフォレストを先に試したほうがいい。こちらのほうが頑健だからである。ランダムフォレストがうまくいったとしても、予測時間が非常に重要な場合や、機械学習モデルから最後の１％まで性能を搾り出したい場合には勾配ブースティングを試してみるとよい。

（4）長所と短所

勾配ブースト法は、教師あり学習の中で最も強力で、広く使われているモデルである。

主な短所はパラメータのチューニングに細心の注意が必要であることと、訓練にかかる時間が長いことである。ほかの決定木ベースのモデルと同じように、特徴量のスケール変換をする必要はなく、２値特徴量と連続特徴量が混在していてもうまく機能する。また、やはり高次元の疎なデータに対してはあまりうまく機能しない。

**参考文献**

・Ｄ－１　SVM（Support Vector Machine）

・Ｄ－２　ロジスティック回帰（Logistic Regression）

https://it-mint.com/2017/08/30/logistic-regression-analysis-1228.html

http://gihyo.jp/dev/serial/01/machine-learning/0018

https://qiita.com/yshi12/items/3dbd336bd9ff7a426ce9

https://analytics-news.jp/info/logi-reg-analysis

・Ｄ－３　k 最近傍法（k-nearest Neighbors）

Python で学ぶ機械学習　2.3.2　k-最近傍法

https://logics-of-blue.com/svm-concept/

・Ｄ－４　ニューラルネットワーク（Neural Network）

Python で学ぶ機械学習　2.3.8　ニューラルネットワーク（ディープラーニング）

Udemy.inc（https://udemy.benesse.co.jp/ai/neural-network.html）

・Ｄ－５　ランダムフォレスト（Random Forest）

Python で学ぶ機械学習　2.3.6　決定木のアンサンブル法

・Ｄ－６　勾配ブースト法（Gradient Boosting）

Python で学ぶ機械学習　2.3.6.2　勾配ブースティング回帰木（勾配ブースティングマシン）

Gradient Boosting について調べたのでまとめる

（http://st-hakky.hatenablog.com/entry/2017/08/08/092031）