デジタルプラクティス　Vol.11 No.1(Jan. 2020)

一般投稿論文

Kaggleカーネルを参照した機械学習アルゴリズムの選択／適用パターンの抽出と習得

晦日 慶太[1]　大内 一哲[2]　岡留 有哉[3]　神崎 元[3]　土屋 俊雄[1]　松岡 賢[4]　吉田 和樹[5]

[1]日本電気　[2]NECソリューションイノベータ　[3]日立製作所　[4]東芝デジタルソリューションズ　[5]国立情報学研究所

近年，IoTデバイスの普及により取得可能なデータが増大し，データを利活用したビジネスが拡大している．データを有効に活用するためには多くのノウハウが必要であり，そのようなノウハウを保有するエンジニアの需要が高まり，人材が不足する状況にある．本研究では，機械学習アルゴリズムの選択／適用パターンを抽出し，スキルが十分でない人材でも適切なアルゴリズムの選択と適用を容易に実現できるようにすることを目指す．実際，抽出したパターンを用いることでアルゴリズム適用に要する時間を約77%削減できることが確認できた．これにより，パターンを用意しておくことで，効率的なアルゴリズム適用が可能になる見通しを得た．

1. はじめに

近年，IoTデバイスの普及により，工場における機器稼働状況データから，個人のバイタル情報まで多くのデータを取得・保存できるようになっている．そのため，これらの多様なデータを活用することで，データからの価値創造によるビジネスが拡大している．データを活用するためにはデータサイエンティストなどのディジタル人材も不可欠であり，これらの人材の需要も急増している．しかしながら，需要の拡大に対し人材の供給は追いついておらず，人手不足が課題になっている[1]．

データを価値に結びつけるためには，データの特性およびビジネスの目的に応じて適切にデータを扱う必要がある．そのような専門知識を獲得するためには，実際にデータを扱うことで得られる知見や，数理的な知識が不可欠である．しかしながら，既存人材のスキルアップやスキルチェンジにより機械学習のスキルを持つエンジニアを増やすことにも多大な時間的コストがかかる．

機械学習の技術を習得するために時間的コストがかかる理由として，数理的基礎知識の習得だけでなく，アルゴリズムの数および種類が多岐にわたることがある．そのため，業務を行う上でどのアルゴリズムを優先して勉強するか，といった優先順位をつけることが容易でない．また，アルゴリズムを習得した場合でも，その利用に至るまでのデータの分析テクニックや，スケーリングなどの前処理に関する知識まで網羅されているとは限らず，適切にアルゴリズムを適用できないという問題もある．

不足する機械学習エンジニアを早期に育成するためには既存のソフトウェアエンジニアやプログラマ等のIT技術者がスキルを身に着け機械学習エンジニアとしても活躍できるようにすることが効果的であると考えている．そこで本稿では，機械学習のスキルが十分でない人材でも効率的なデータの活用を可能とするために，データサイエンスにおける代表的なコンペティションであるKaggle[2]から，アルゴリズムの適用だけでなく，データの前処理を行う上で必要なノウハウ等も含めた機械学習アルゴリズムの適用パターン（以後，「適用パターン」と呼ぶ）の抽出を行った．また，それと併せて，データの量や質的変数の数などの特性を見た上で，どのアルゴリズムを適用すればよいかを示した機械学習アルゴリズムの選択パターン（以後，「選択パターン」と呼ぶ）の抽出も行った．

これらにより，新たなデータを得た際に，どのアルゴリズムを選択すればよいかが確認でき，それが未知のアルゴリズムであれば，適用例とともに習得できるようになる．

アルゴリズムの効果的な適用だけを目的とするならば，すべてのアルゴリズムを並列に動かして一番精度が高いものを選択する DataRobot[11] など，自動化に向かうアプローチもある．しかし，本研究では，人材育成に主眼を置き，機械学習のスキルが十分でない人材に，アルゴリズムを習得してもらえるように，パターンを記述・利用するアプローチを取った．

本稿では，適用パターンがあることで，機械学習の実装時間がどれほど短縮できるかを実験により確認した結果を報告する．選択パターンについても，同様に，それがあることで，機械学習アルゴリズムの選択をどれほど正確かつ効率的に行えるかを確認する必要があるが，それには実験内容や実施方法の検討を重ねる必要があったため，今後の課題とする．

2. 事前準備

本稿は，国立情報学研究所（以後，NIIと呼ぶ）の社会人向け教育プログラムであるトップエスイー[5]のソフトウェア開発実践演習での活動をもとに執筆した．

本章では，選択／適用パターンについて，抽出の対象とする機械学習アルゴリズム，パターンを構成する項目，パターンの抽出プロセスを説明する．

2.1 パターンの抽出対象アルゴリズム

Kaggleとはデータサイエンスや機械学習に関心のある人たちが集うコミュニティサイトである（2017年時点で登録ユーザ数100万人以上）．Kaggleの特徴は企業や政府がデータ分析の課題とデータを提示し，最も精度の高い分析モデルを提出した参加者に賞金を支払うコンペティションが毎日開催されていることである．初学者向けにはカーネルと呼ばれる機能があり，そこでは他のユーザがコンペティションで構築した予測モデルとそのコードの説明が公開されている．Kaggleではコンペティションごとに議論用および参考用にこのカーネルが設けられている．さらに，これらのカーネルの中から特に参考になるものを選び出して，アルゴリズムごとにまとめたカーネルも投稿されている．

本研究では，上記のアルゴリズムごとにまとめたカーネルの1つであるData Science Glossary on Kaggle[3]をもとにして，そこで挙げられたアルゴリズムの中から，主に，初学者にとっての重要度，難易度，活用場面の観点で絞り込みを行い，以下の14種類のアルゴリズムを選定し，適用パターンの抽出を行った．

■教師あり学習
1. 線形回帰
2. Ridge回帰
3. Lasso回帰
4. Logistic回帰
5. Decistion Tree
6. Random Forest（分類）
7. Random Forest（回帰）
8. Gradient Boosting Tree（LightGBM実装）
9. Support Vector Machine（分類）
（以後，「SVC」と呼ぶ）
10. Support Vector Machine（回帰）
（以後，「SVR」と呼ぶ）
11. k-Nearest Neighbor
（以後，「kNN」と呼ぶ）

■教師なし学習
12. K-means Clustering
13. Hierarchical Clustering
14. DBSCAN

ほかにも，深層学習を含め多数のアルゴリズムが存在するが，それらからのパターンの抽出は，引き続き行っていく予定である．

本稿では，第4章において，上記のアルゴリズムの1つであるLasso回帰の適用パターンを示す．

2.2　パターンの抽出プロセス

NIIのソフトウェア開発実践演習のスケジュールを下の**図1**に示す．

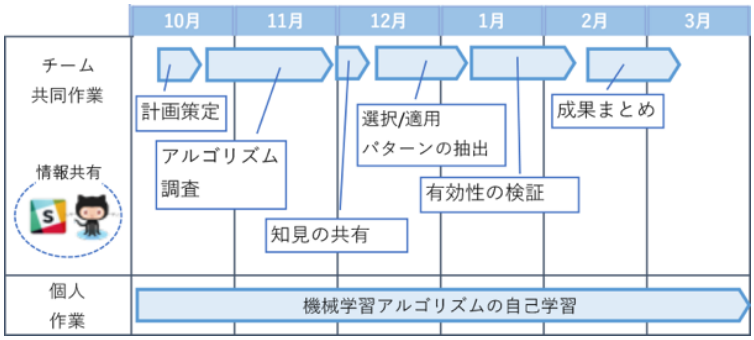


図1　演習スケジュール

演習はチームで行われ，このチームは各企業から派遣された6名のメンバと1名の講師により編成された．さらに，6名のメンバのうち，1名は機械学習の熟練者であり，残り5名は初学者であった．

以下に，パターンの抽出プロセスを時系列で説明する．

2.2.1　計画策定

最初は計画策定フェーズとして，おおむね隔週の頻度でメンバ全員が集合し，演習目標，課題設定，スケジュールのすり合わせを行った．

2.2.2　アルゴリズム調査

先述の14種類のアルゴリズムについて，各メンバが2種類ずつ調査を行い，その結果を他のメンバに説明し，調査を通して得られた知見を共有した．なお，共有の際には，3人1組で2グループに分かれ，各グループが交互に隔週で集合することで，十分な調査期間の確保とメンバ全員に適度な負荷がかかるように，調整が図られた．

2.2.3　得られた知見の共有

アルゴリズムの調査が完了し，グループ内で共有された知見を，さらにグループ間でも共有した．これにより，メンバは，アルゴリズムごとに，それがどのような性質のデータや問題に向くのかの理解に加え，アルゴリズム間の関係性（類似性や相反性など）や，Pythonによる実装時の注意点などにも理解を深めることができた．

この知見の共有を通して得られたアルゴリズム間の関係性に対する理解は，選択パターンを抽出する上で，必要不可欠なものとなった．

2.2.4　選択パターンと適用パターンの抽出

知見の共有の後，選択パターンと適用パターンを抽出した．

（1）選択パターンの抽出

与えられたデータの特徴に対してどのアルゴリズムを選択すればよいかという観点で，チーム内で議論を重ね，第3章に示すフローチャート形式の選択パターンを抽出した．

（2）適用パターンの抽出

適用パターンの抽出では，オブジェクト指向設計において，広く普及しているデザインパターン[4]を参考にして，パターンを構成する項目をテンプレートとして定め，これにしたがって，抽出した結果を記述した．具体的には，機械学習アルゴリズムの適用に際して，理解しておくべき点を踏まえ，パターンを構成する項目として以下を独自に選定した☆1．

- パターン名と分類
- 目的
- 課題例（Kaggle上の問題／データセット）
- 適用条件
- 適用手順
- 実装上の注意点（Python3）
- サンプルコード（Jupyter Notebook形式）
- 適用結果
- 理論的背景
- 出典および参考文献
- 関連するパターン

実装上の注意点やサンプルコードについては，Jupyter Notebook上でPython3を用いた例を載せている．

2.2.5　選択パターンと適用パターンの有効性検証

両パターンの有効性に関して，次の2点の仮説を立てた．

（1） 選択パターンを活用することで，データの特徴（データ量，変数の数など）に合致した適切なアルゴリズムが選択できる．
（2） 適用パターンを活用することでソースコードの実装時間が短縮する．

仮説検証プロセスのフィジビリティについて検討した結果，仮説（1）はチーム以外の被験者によるABテストが必須であり，残された演習の期間で実施するには厳しいことが分かった．そこで，仮説（1）は今後の課題とした．

3. 選択パターン

アルゴリズムの調査を通して，各アルゴリズムには特性があり，データによって適用すべきアルゴリズムは異なり，適切に選択する必要があることが分かった．従来は，データに合わせた適切なアルゴリズムを機械学習スキルの高い人材が持つノウハウにより選択していた．そのため，スキルが十分でない人材が適切なアルゴリズムを選択することは難しく，十分なデータ分析を行うことができなかった．そこで共有された知見をもとに，スキルが十分でない人材であっても容易に適切なアルゴリズムが選択可能になる選択パターン（**図2**）を抽出した．

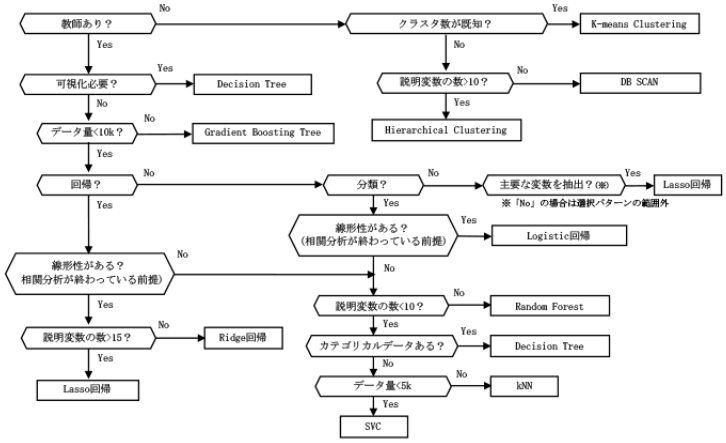


図2　選択パターン

まず，データ自体が教師ありか教師なしかで分類する．教師なし学習では，クラスタリング問題となり，クラスタ数があらかじめ決まっているかどうかで分岐する．クラスタ数が既知の場合，クラスタ数が入力として必要になるK-means Clusteringを使用する．クラスタ数が未知で説明変数が少ない場合に限り，DBSCANを使用する．これは，DBSCANが，入力としてクラスタ数が不要であり，説明変数が多くなると計算量も多くなるためである．一方，説明変数が多い場合はHierarchical Clusteringを使用する．

教師あり学習では，回帰問題または分類問題となる．これらの問題に対して，予測結果を可視化するかどうかによって使用するアルゴリズムの方向性が決まる．予測結果の過程を可視化する必要がある場合，説明変数の値から予測結果へのツリー分岐により可視化が可能なDecision

Treeを使用する．可視化が不要で，データ量が多い場合には，多量のデータに対して短時間で分析が可能なGradient Boosting Treeを使用する．一方，Gradient Boosting Treeはデータ量が少ないと過学習を起こしてしまい十分な性能を達成できないことから，データ量が少ない場合には，他のアルゴリズムを使用する必要がある．

データ量が少ない回帰問題では，線形性があるかどうかを確認する．アルゴリズム選択パターンにおける線形性があるとは，目的変数と説明変数の間でピアソンの相関係数を求め，0.8以上であるものとする．相関分析等で線形性があるかどうかを確認し，線形性があると想定される場合，線形回帰アルゴリズムを使用する．説明変数が多い場合は，次元削減効果のあるLasso回帰を使用する．一方，説明変数が多くない場合は，すべての説明変数を用いても過学習の影響が過大にならないことが期待できるRidge回帰を使用する．

また，分類問題で線形性がある場合，計算量の少ないLogistic回帰を用いる．

回帰問題，分類問題において線形性がないと想定される場合は，非線形問題に対応可能なアルゴリズムを選択する．説明変数の数が多い場合は，Random Forestを用いる．説明変数の数が少なく，カテゴリカルデータがある場合は，カテゴリカルデータをそのまま処理可能なDecision Treeを使用する．説明変数が少なく，データ量も少ない場合は，多様な非線形データに対応可能なSVCを使用する．しかし，SVCはデータ量が多くなると計算量が大幅に増加するため，データ量が多い場合はkNNを使用する．

データ分析の目的が，回帰でも分類でもなく，データに影響する主要な説明変数を抽出する場合には，次元削減効果のあるLasso回帰を利用して抽出を行う．

4. 適用パターン

本章では，選定した14種類のアルゴリズムの1つであるLasso回帰について，適用パターンに記載した内容を簡潔に紹介する．

例：Lasso回帰

Lasso回帰は一次のペナルティ項（L1ノルム）を付加した線形回帰のアルゴリズムである．L1ノルムを付加することで係数パラメータがゼロになることが多くある．この特徴を利用して，Lasso回帰は単に線形回帰の問題を解くだけでなく，どの特徴量が重要であるかを抽出することができる．

以下に，第2章で説明したテンプレートに従って，Lasso回帰について記述した例を示す．

- パターン名と分類
  パターン名：Lasso回帰の適用パターン
  分類：「回帰／特徴量抽出」
- 目的
  ・既知の説明変数から未知の量的変数を予測する
  ・説明変数の中で重要なものを抽出する
- 課題例
  House Prices：Advanced Regression Techniques [6]アイオワ州の住宅価格を79の説明変数から予測する（回帰）
- 適用条件
  1. 目的変数のヒストグラムがひと山になること

2. 特徴量が多く，重要なものがわずかしかないと予想されること
3. （伝統的に）線形の関係になると予想されること

- 適用手順

図3は回帰問題を解くためのLasso回帰の適用手順である．Lasso回帰で特徴量抽出を行う場合，（7）の「最適値を使ってLasso回帰を適用」の後に回帰係数がゼロのものがあれば，その説明変数を削除する．その後，Lasso回帰のペナルティ項の効果を調整するハイパーパラメータ☆2の最適値を求め，Lasso回帰を適用する．回帰係数がゼロになるものがなくなるまでこれを繰り返し行うことで特徴量抽出ができる．

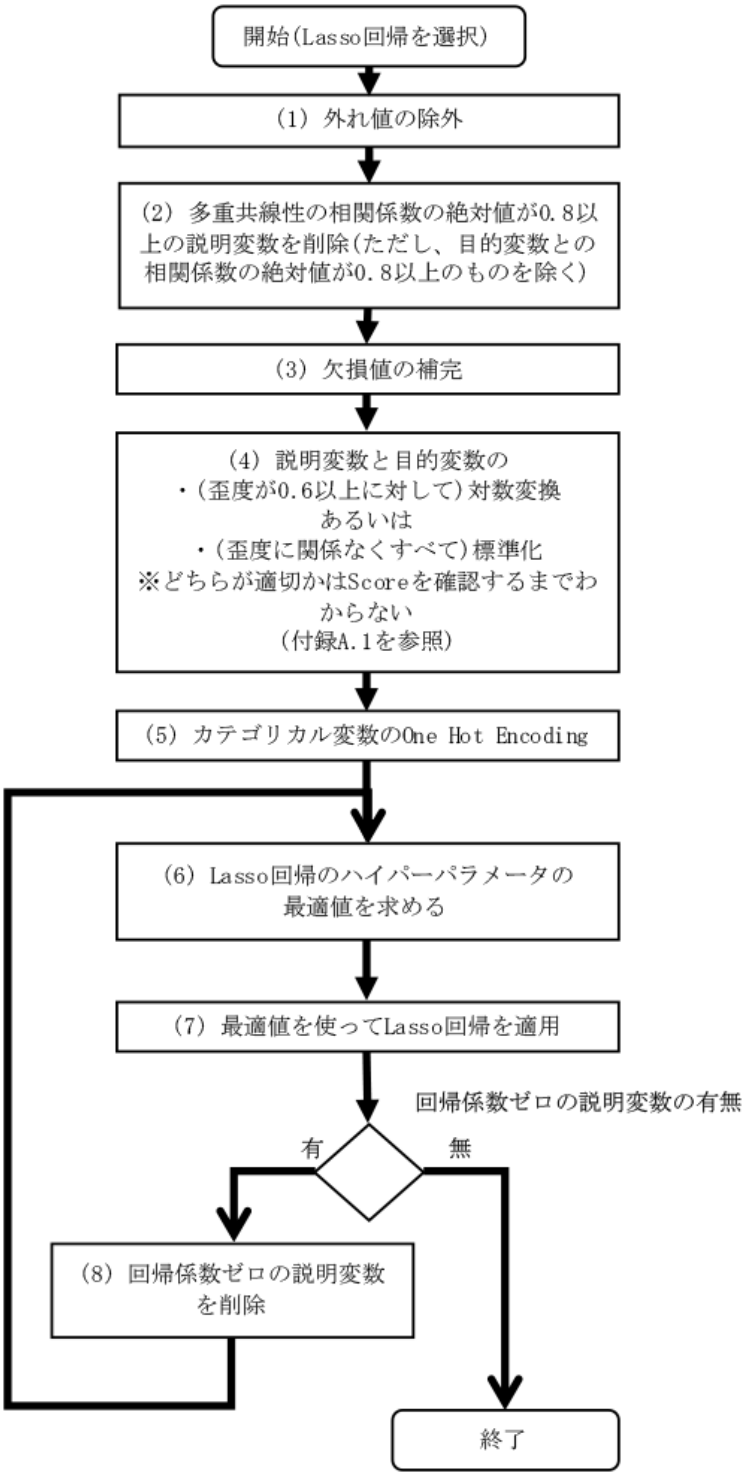


図3　Lasso回帰の適用手順

適用手順の（2）に記した相関係数の絶対値0.8は，それ以上であれば強い相関関係があると一般的に言われる値を固定で採用した．（4）に記した歪度0.6についても，一般に使われている値を固定で採用した．

適用手順の（4）の内容（対数変換，あるいは，標準化）は対象データにより異なる．現時点では，対象データの特徴からどちらを選択すべきかを完全には特定できていないため，今後の課題とする（付録A.1を参照）．

- 実装上の注意点（Python）
  対数変換した場合はLasso回帰の予測値のexpをとる．
- サンプルコード（Jupyter Notebook形式）
  データの読み込みから特徴量抽出までを行うサンプルを用意（付録A.2を参照）．
- 適用結果
  サンプルコードの実行結果について記載（付録A.3を参照）．
- 理論的背景
  L1正則化[7]することでスパースな解を得られやすい．
- 出典および参考文献
  （本稿では割愛）
- 関連するパターン
  Rigde回帰：Lasso回帰がL1正則化なのに対してRidge回帰はL2正則化[8] 線形回帰：正則化項がない回帰モデル

5. 評価

5.1 実験

適用パターンを利用することにより，初めて見るデータセットに対して，機械学習の実装時間が短縮できることを，実験を通して検証する．データセットはUCI Machine Learning Repository[9]から選んだものを利用した．このリポジトリに記されたデータセットに対する概要説明をもとに，その特徴に応じて，適用するアルゴリズムと達成すべき予測精度（平均二乗誤差，正解率）の目標値をあらかじめ定めておく．そして，チームのメンバを被験者として，データセットを割り振り，被験者が機械学習の実装に要した時間を計測し，また，実装したモデルの予測精度が目標値に到達するか否かを確認した．実装時間については，適用パターンの抽出前に，調査の段階で実装に要した時間と比較することで，適用パターンの有無による実装時間の差異とした．なお，アルゴリズムにより計算時間が異なるため，プログラムの実行時間は実装時間に含まないこととする．

実験の進め方に関する具体的なプロセスを以下に示す．

- メンバは，各自がすでに調査したアルゴリズムを対象に，問題と模範解答（ソースコード）を作成し，講師に提出する．
- 講師は，メンバに問題を割り振る．その際，メンバには，調査を担当したアルゴリズムから，できるだけ遠い関係にあるアルゴリズムを割り振るように留意した．
- メンバは，講師が指定した問題を解き，ソースコードの実装時間と実装したモデルの予測精度とを併せて，解答（ソースコード）を講師に提出する．
- 講師は，解答をメンバに公開する．
- 模範解答を作成したメンバは，解答を評価し，解答者にフィードバックする．

5.2 結果

図4は，被験者ごとに，適用パターンを参照していない状態（以後，「Before」と呼ぶ）と参照した状態（以後，「After」と呼ぶ）での実装時間を示したものである．1人の被験者がBeforeとAfterで使用したアルゴリズムは異なるが，その実装時間を比較することにより，任意のアルゴリズムについて，適用パターンの参照の有無が，実装時間に与える影響を確認することができる．ここでは，それまでの機械学習の適用経験による影響を排除するために，初学者の被験者のみを対象とした．このような前提のもと，実際に計測した結果を比べると，適用パターンの参照により，最大95%，平均77%の実装時間を削減できたことが分かった．

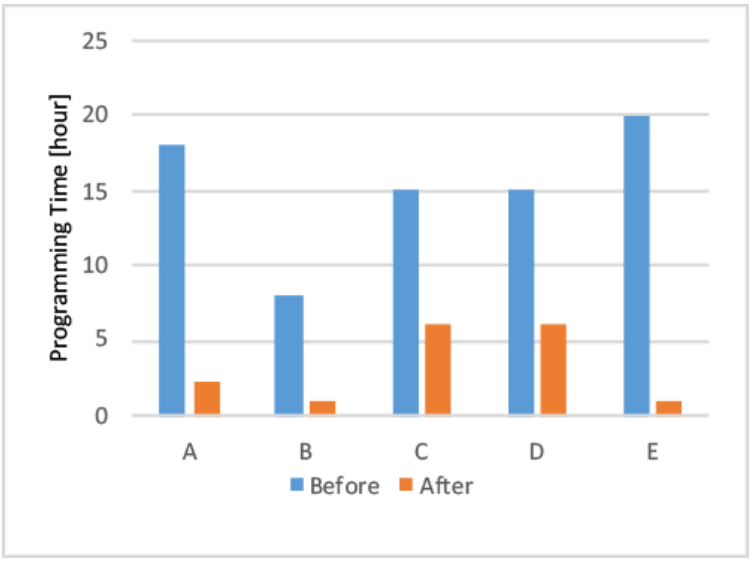


図4　被験者（初学者）ごとの実装時間の比較

また，BeforeとAfterで各被験者が使用したアルゴリズムが異なるため，**図5**にアルゴリズムごとに集計した実装時間で比較した結果を示す☆3．Logistic回帰，Lasso回帰，Decision Treeについては，BeforeとAfterで，ともに初学者が要した時間を記録しており，アルゴリズムごとに見ても，大幅な削減効果が認められる．Gradient Boosting Treeは，Beforeが初学者でAfterが熟練者であることから，前記3つのアルゴリズムに対する削減効果は，初学者と熟練者の間の時間差を埋めるほどであったと見ることもできる．さらに，Random Forest（回帰）とSVCについては，Beforeが熟練者でAfterが初学者であるため，適用パターンを参照することで，初学者でも熟練者とほぼ同じレベルにまで実装時間を短縮できることが分かる．

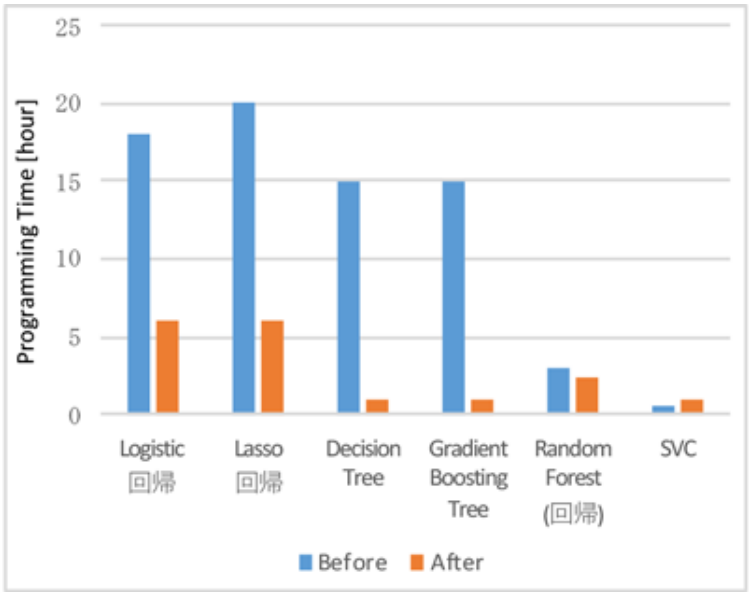


図5　アルゴリズムごとの実装時間の比較
（熟練者はSVCのBeforeとRandom Forest（回帰）のBefore, Gradient Boosting TreeのAfterに該当する）

表1は，今回の実験で割り振ったアルゴリズムについて，予測精度に関する目標値と解答値を並べて示したものである．適用パターンを参照することで，精度においても，目標値を超える結果が得られていることが分かる．

表1　アルゴリズムごとの予測精度の比較（抜粋）

アルゴリズム	予測精度(※1)	
	目標値(※2)	解答値
Logistic回帰 (テキスト分析)	0.996以上	0.97
Lasso回帰	10未満	3.00
Random Forest(回帰)	1.0未満	0.996(※3)
SVC	0.98以上	1.00
Decision Tree	0.9以上	0.94
Gradient Boosting Tree (LightGBM実装)	0.57以上	0.60

（※1）回帰の場合は平均二乗誤差，分類の場合は正解率
（※2）目標値はアルゴリズムの調査担当者の実績値よりも少し低め（実績値に対し数%～10%ポイント下げる）に設定
（※3）目標値と比較するため小数第3位まで表示

6. 考察

6.1 適用パターンによる効果に関する考察

機械学習の実装時間は，データの分析も含めた前処理とアルゴリズム適用に分けることができ，一般に8割は前処理であるという経験則がある[10]．適用パターンには，データの前処理も含めたノウハウを記載しているため，前処理においても時間短縮効果があり，結果として大きな時間短縮が可能になったと考えられる．

一方，今回の実験では，BeforeとAfterの間で，2.3節の抽出プロセスに沿った流れで，個人で試行錯誤を繰り返しながらプログラミングのスキルを高め，また，チーム内での情報共有から機械学習の知識を習得することも行われていたため，時間短縮には適用パターン以外の効果も含まれていると考えられる．**表2**に，実装時間に影響を与える要因と，パターン適用による効果，および，抽出プロセスを通じて得られた効果の関係を定性的に示す．適用パターンには，プログラミングスキルに関するものや，環境構築・設定等のノウハウは含まれていない．これらについては，抽出プロセスを通して得られたと考えられる．

表2　実装時間に影響を与える要因と効果の関係

	パターンの効果	抽出プロセスの効果
心理的障壁	△	○
機械学習の知識	○	○
Pythonスキル	△	○
ライブラリ知識	△	△
プログラミング環境設定	×	○
前処理	○	○
アルゴリズム適用	○	○
性能評価	○	○

本実験では，適用パターンを抽出したメンバ同士が被験者になり，クロスチェックにより効果を確認した．しかし，適用パターンの効果を正確に評価するためには，抽出プロセスにかかわらなかった被験者を集めて，パターンを参照するグループと参照しないグループに分け，双方の実装時間の差を評価する必要があると考える．

適用パターンの効果の正確な評価については，今後の課題である．

6.2 機械学習の習熟時間に関する考察

チームのメンバのうち5名はPythonおよびデータ分析の初学者であったにもかかわらず，初めて見るデータセットに対して短い時間で機械学習アルゴリズムを適用して結果を得られるようになった．多くの参加者は他のプログラミング言語には習熟していたがPythonにはほとんど触れたことがない状況であった．その中で，各自が2種類のアルゴリズムについて適用パターンを抽

出し，また，チーム内での議論を通して選択パターンを抽出することで，実験では実装時間を短縮することができた．このことから，ある程度アルゴリズムを定めた上で，機械学習の実装に習熟しようとすると，抽出プロセスの各アクティビティに初学者が費やした平均時間（**表3**を参照）から，全体でおよそ50時間を必要とすることが分かった．

表3　抽出プロセスの各アクティビティに費やした時間

アクティビティ	初学者が費やした平均時間(H)
アルゴリズム調査	30
得られた知見の共有	4
選択パターンと適用パターンの記述	8
選択パターンと適用パターンの有効性検証	8
合計	50

今後は，基礎的な数学的原理の理解を含めた習熟時間の推計を行うことで，企業の人材育成に活かすことができると期待される．

6.3　機械学習のためのチートシートとの連携に関する考察

機械学習においてよく知られているチートシートとして，

① ニューラルネットワークや機械学習さらにPythonを使ったデータサイエンスについてまとめたCheat Sheets for AI[12]
② scikit-learnのアルゴリズムチートシート[13]
③ クラウドサービス上での機械学習についてまとめたAzure Machine Learning Studioの機械学習アルゴリズムチートシート[14]

等がある．

①は，特定のライブラリを使って何らかの処理を実行したいときの一連のコード群を，逆引きインデックスのような形でまとめた内容になっている．これは，機械学習アルゴリズムの習得のために利用するというよりも，習得が済み，業務への実用段階に入った人材が，実装時に自身の記憶を補うために利用するものと考えられる．また，②と③は，scikit-learnやAzureといったライブラリやクラウドサービスに特化して，アルゴリズムを選択するためのフローチャートが示されており，役割としては，選択パターンに一致するが，環境が決まっているという点で，①と同様に実用に近い段階で利用するものと捉えられる．

したがって，パターンとチートシートは，それぞれ使われる段階や状況が異なっているため，互いに競合する関係にあるものではなく，人材の成長に合わせて，パターンからチートシートに移行していく形で両者を連携させることができれば，ともに機械学習アルゴリズムの適用にとって有益な再利用資産になり得ると考える．

7. まとめと今後の課題

本研究では，機械学習のスキルが十分でない人材が，容易にデータを活用できるようにすることを可能にし，機械学習に熟練した人材の不足に対処していくことを目的に，次の2種類のパターン，すなわち，データの特徴に合致した適切な機械学習アルゴリズムを選択できるようにする選択パターンと，機械学習のアルゴリズムごとに，それを適用するために必要なノウハウをまとめた適用パターンを抽出した．

また，実験を通して，適用パターンを利用することで，機械学習の実装時間を平均で77%削減することができ，適用パターンによりスキルが十分でない人材でも効率的に機械学習アルゴリズムを適用することが可能となる見通しを得た．

ただし，この実装時間の短縮効果が，適用パターンの効果によるものか，あるいは，パターンの抽出プロセスを通して獲得したプログラミングのスキルや，チーム内での情報共有で習得した機械学習の知識によるものか現時点では判断することができない．

そこで，今後は，適用パターンの効果と選択パターンの効果を併せて検証するために，抽出プロセスにかかわりがない第三者の被験者に対して，パターンを参照するグループと参照しないグループに分けて，効果を判定するための実験を企画・実施していく予定である．

さらに，複数のアルゴリズムを組み合わせて適用することで，予測精度が向上することが，一般に知られている（これはアンサンブル手法と呼ばれる）．パターン抽出の対象範囲を，アンサンブル手法にまで拡げる場合には，選択パターンの表現形式であるフローチャートの中で，複数のアルゴリズムの組合せも含めて，分岐条件等を考えなければならなくなる．そのためには，どのような条件で，その組合せが最良の結果を生むのかを明確にしておく必要があり，それも今後の課題である．

また，今回抽出した選択／適用パターンを実際のデータ分析業務を行っているチームや個人に提供し，利用してもらい，その内容や対象としているアルゴリズムの充実度などに関する意見を収集し，改善することも重要となる．今回はKaggleのカーネルをもとにして，多くの場面で利用される代表的なアルゴリズムを対象にしてパターンの抽出を行ったため，実業務において重要となるデータ加工などの処理が抜けている可能性がある．また，Kaggleでは取り上げられることが多くない時系列分析などの課題を，実業務では扱うことも多い．Kaggleと併せて，このような現場の意見やノウハウを取り込み，パターンを改善していくことも今後の課題である．

参考文献

1）前　一平：AI時代を担う人材の育成，立法と調査 2018.10, No.425, pp.48-49 (2018).
2）Kaggle : https://www.kaggle.com/
3）Kaggle : Data Science Glossary on Kaggle, https://www.kaggle.com/shivamb/data-science-glossary-on-kaggle
4）エリック ガンマ 他：オブジェクト指向言語における再利用のためのデザインパターン，ソフトバンククリエイティブ（1999）．
5）トップエスイー：https://www.topse.jp/ja/
6）House Prices : Advanced Regression Techniques, https://www.kaggle.com/c/house-prices-advanced-regression-techniques
7）Tibshirani, R. : Regression Shrinkage and Selection via The Lasso, Journal of The Royal Statistical Society, Series B (Statistical Methodology), Vol.58, No.1, pp.267-288 (1996).
8）Hoerl, A. E. and Kennard, W. R. : Ridge Regression : Biased Estimation for Nonorthogonal Problems, Technometrics 12 (1970).

9）UCI Machine Learning Repository : https://archive.ics.uci.edu/
10）有賀康顕 他：仕事ではじめる機械学習，オライリージャパン（2018）．
11）DataRobot : https://www.datarobot.com/jp/
12）Cheat Sheets for AI : Neural Networks, Machine Learning, Deep Learning & Big Data, https://becominghuman.ai/cheat-sheets-for-ai-neuralnetworks-machine-learning-deep-learning-big-data-678c51b4b463
13）Choosing The Right Estimator : https://scikit-learn.org/stable/tutorial/machine_learning_map/
14）Azure Machine Learning Studio の機械学習アルゴリズム チート シート : https://docs.microsoft.com/ja-jp/azure/machine-learning/studio/algorithm-cheat-sheet

脚注

☆1　5.1節で説明する実験の後，メンバの協議により，パターンの利用しやすさの観点から，この項目の中の“課題例”と“適用結果”については，“サンプルコード”の中にまとめて書くことになった．
☆2　ハイパーパラメータとは，予測の枠組みの中では決定されず，利用者が設定するパラメータのこと．たとえば，学習率，バッチサイズ，学習イテレーション数，など．
☆3　図4のBのBeforeの時間の計測対象となったアルゴリズム（Ridge回帰）は，実験で誰にも割り振られなかったものである．そのため，図5には，この値は現れていない．

付録

A1.

Lasso回帰の適用手順（4）の，対数変換すべきか，あるいは，標準化すべきかは，データにより異なることが分かったが，その条件を明らかにできていない．

一般に回帰分析における説明変数の標準化は，数値データについて単位がまちまちであるため変数どうしを比較可能にするために必要な処理である．また，歪度が高い（0.6以上）変数を対数変換するのは，歪度が高いと分布の山が下に偏っているのを対数変換することで分布を正規分布に近づくように矯正するためである．回帰分析では変数が正規分布することが前提になっているため，分布を矯正する対数変換は必要な処理である．

したがって，標準化と対数変換のどちらも必要な処理と考えるが，実際のデータに適用するとどちらか一方を適用した方が精度が良くなった．その理由については，次の2つのデータ特性が関係しているのではないかと考える．

（1）欠損値が多く含まれるデータ

欠損値が含まれる場合，標準化前に欠損値の穴埋めをする必要がある．通常，平均や中央値，最頻値といった値で一律に穴埋めをする．欠損値が多く含まれるデータの場合，穴埋めをしたデータが多いため本来の分布から乖離して標準化がうまくできないことが考えられる．したがっ

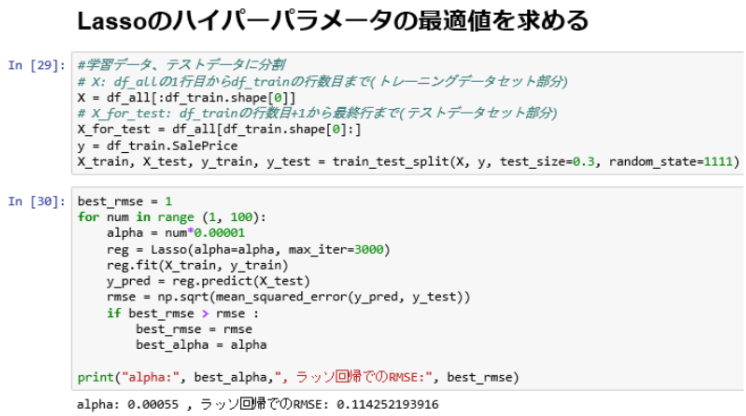
て，欠損値が多く含まれるデータについては，標準化はしないほうがよいと考える．

（2）目的変数がゼロ付近の値を多く持つデータ

ゼロ付近の値の対数をとる（log1p）と，元の値のわずかな差が，対数変換後の大きな値の差になる．この差がモデル構築に影響し，精度が得られなくなってしまうことが考えられる．したがって，目的変数がゼロ付近の値を多く持つデータについては，対数変換しない方がよいと考える．

A2.

第4章のLasso回帰のサンプルコードの抜粋を以下の**図6**に示す．



Lassoのハイパーパラメータの最適値を求める

```
In [29]: #学習データ、テストデータに分割
         # X: df_allの1行目からdf_trainの行数目まで(トレーニングデータセット部分)
         X = df_all[:df_train.shape[0]]
         # X_for_test: df_trainの行数目+1から最終行まで(テストデータセット部分)
         X_for_test = df_all[df_train.shape[0]:]
         y = df_train.SalePrice
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1111)

In [30]: best_rmse = 1
         for num in range (1, 100):
             alpha = num*0.00001
             reg = Lasso(alpha=alpha, max_iter=3000)
             reg.fit(X_train, y_train)
             y_pred = reg.predict(X_test)
             rmse = np.sqrt(mean_squared_error(y_pred, y_test))
             if best_rmse > rmse :
                 best_rmse = rmse
                 best_alpha = alpha

         print("alpha:", best_alpha,", ラッソ回帰でのRMSE:", best_rmse)

         alpha: 0.00055 , ラッソ回帰でのRMSE: 0.114252193916
```

図6　サンプルコード例

実際のサンプルコードにはデータの読み込みや外れ値の除外等の前処理も含む．

A3.

第4章のLasso回帰の適用パターンの中で，Lasso回帰を繰り返し適用することで特徴量抽出ができることを紹介した．実際に，第4章の問題でLasso回帰を繰り返し適用した結果が以下の**表4**である．表4の説明変数の数はカテゴリカル変数をOne Hot表現に変換した後の数である．この結果を見ると，1回目の適用で190もの説明変数の回帰係数がゼロになっている．一方，2回目の適用では1つしか回帰係数がゼロになっていない．このことから，この問題では，Lasso回帰を1回適用することで有効な説明変数をほぼ特定できたと考える．また，機械学習による予測値とコンペティションの正解値の二乗平均平方根誤差（RMSE）の値は各回で大きな差はなかった．

表4　Lasso回帰適用時の説明変数の数とRMSE

適用回数	説明変数	RMSE	Kaggle順位
1	283	0.11703	614位
2	93	0.11708	622位
3	92	0.11703	614位

晦日 慶太（非会員）k-misoka@cb.jp.nec.com
2014年上智大学大学院理工学研究科物理学専攻博士課程（前期）修了．同年日本電気（株）入社．以来，製造・装置業システム本部にてPLMシステムのSIに従事．

大内 一哲（非会員）kaz-oouchi@sx.jp.nec.com
1997年電気通信大学電子工学科卒業．同年新日本無線（株）に入社．ディジタル信号処理の研究開発を経て2007年にNECソフト（現：NECソリューションイノベータ）（株）に入社．組み込み系のSIを経て，現在は社内のプロジェクト活動データの分析業務に従事．

岡留 有哉（非会員）yuya.okadome.qj@hitachi.com
2016年大阪大学大学院基礎工学研究科博士課程修了．同年，（株）日立製作所　研究開発グループ入社．現在に至る．機械学習，データ分析，ロボティクスおよびシステム知能化の研究に興味を持つ．博士（工学）．

神崎 元（非会員）hajime.kanzaki.ad@hitachi.com
2007年京都大学大学院通信情報システム専攻博士課程（前期）修了．（株）日立製作所入社．以来，研究開発グループにて無線通信システムの研究に従事．IEEE会員．

土屋 俊雄（非会員）t-tsuchiya@bk.jp.nec.com
2006年東京理科大学理学研究科数学専攻博士課程（前期）修了．同年日本電気（株）入社．現在は大規模のアプリケーション開発に従事．

松岡 賢（非会員）Matsuoka.Masashi@toshiba-sol.co.jp
2002年横浜国立大学大学院工学研究科人工環境システム学専攻博士課程（前期）修了．同年（株）東芝に入社．2004年東芝ソリューション（株）（現：東芝デジタルソリューションズ（株））に転籍．現在に至る．現在は情報システム開発技術の開発，普及展開業務に従事．

吉田 和樹（非会員）k-yoshida@nii.ac.jp
1989年東京工業大学大学院総合理工学研究科システム科学専攻博士課程（前期）修了．同年（株）東芝入社．ソフトウェア再利用のための研究・開発・普及活動に従事．1995～97年イリノイ大学客員研究員，2005年東京工業大学大学院社会理工学研究科経営工学専攻社会人博士課程修了，2007～08年東京大学非常勤講師，2018年より国立情報学研究所GRACEセンター特任研究員．博士（工学）．

2019年4月15日
投稿受付：2019年4月15日
採録決定：2019年9月20日
編集担当：浦本 直彦（（株）三菱ケミカルホールディングス）