機械学習を用いた授業クラス分け自動化

高　瀬　　剛

要　旨

2020 年の新型コロナウィルス禍の中、社会活動・経済活動の維持において多くの面でディジタル技術が貢献した。これはDX（ディジタルトランスフォーメーション）と呼ばれる、ディジタル技術を社会構造の中核に据えて我々の生活・労働の効率・品質を高める取り組みの成果の一端である。現在の日本では、各所でDX、すなわち身近なモノやコトをディジタル技術で高度化することを推進しており、このDXの取り組みはSociety5.0 の土台となるものである。

本稿ではDXの一例として、機械学習を用いた高等教育機関での授業クラス分けの自動化を試みた。今回試行した学習モデルとデータセットの組み合わせおよび機械学習の正答率の議論を通じて、機械学習による授業クラス分けの自動化の問題を考察した。

キーワード：機械学習、Society5.0、DX、教師あり学習、授業クラス分け

1. はじめに

新型コロナウィルスCOVID-19 が世界的に流行する 2020 年の状況下において、社会活動・経済活動の維持のためディジタル技術が多くの面で貢献したことは記憶に新しい。このディジタル技術の貢献はDX（ディジタルトランスフォーメーション）と呼ばれる「ディジタル技術を社会構造の中核に据えて我々の生活・労働の効率・品質を高める取り組み」の成果の一端でもある。DXは「働き方改革」の礎となり、また高度情報化社会の先にあるSociety5.0 に結びついていくものである。

近年の機械学習・深層学習をはじめとする人工知能技術の研究および応用の飛躍的な発展が、DXと呼ばれるディジタル技術の進展に大きく寄与している。最近では、機械学習・深層学習を応用するためのプラットホームが整備されてきており、必ずしも人工知能の専門家でなくても機械学習の訓練データさえ用意すれば分類問題や回帰問題等に対しては比較的容易に適用してみることが可能となっている。

著者の身近な分類問題の一例としては、試験の結果による授業のクラス分けが挙げられる。大

学などの高等教育機関でも、学修効果を最大限にするなどの理由で適切な規模でクラス分けを行うことがある。このクラス分けの作業では、多くの場合、試験の成績順に並べて各クラスが均等な人数になるように配分するか、あるいは試験の成績に加えて学生の普段の様子や授業に対するモチベーションなどを考慮して配分するなどしていると推察する。これらのクラス分けの作業は一般に表計算ソフトウェアなどを用いて手作業で行うと考えられるが、これらの作業は学期ごとに行う必要があり、また処理のための時間も限られることからとても煩雑な作業である。また、試験結果などの客観的な数値ではない要素でクラス分けを行う場合には、作業の担当者の主観によってクラス分けの結果に差が出てくることがあるだろう。

このような身近で煩雑な作業に対して機械学習を適用することで作業の効率化ができるとすれば、無駄な労働の抑制になるばかりでなく、このような機械学習の適用事例を積み重ねていくことでDXを推進することになり、それらはSociety 5.0 の礎となるはずである。本稿では、手軽に利用可能なオープンソース機械学習ライブラリを用いて、高等教育機関での授業のクラス分けの問題に対して機械学習の手法を応用すること、さらにこの過程を通じてクラス分けの自動化の可能性について検証していくことを目的とする。

2. 機械学習の概要

本稿で利用する機械学習についての概要を説明する。機械学習は人工知能研究の一分野として理解されており、研究の進展により機械学習が広く応用されるようになってきている[1]。また、機械学習の一部分としてニューラルネットを用いた深層学習と呼ばれる領域がある。そもそも機械学習、すなわちmachine learningという言葉は 1959 年にArthur Samuel により初めて使われたとされる［2］［3］。その際の定義は “Field of study that gives computers the ability to learn without being explicitly programmed” と表されている。これは「明示的にプログラムすることなしに学習する能力を計算機に与える研究分野」というような意味である［4］。

現状での機械学習は、多数のデータから特徴的な量を抽出して、データの分類、回帰処理、クラスタリング（データのグループ分け）などを行うものを指している［5］。機械学習を大別すると、「教師あり学習」、「教師なし学習」、「強化学習」などがある。例えば出力データ（教師データ）が与えられる「教師あり学習」は分類や回帰などに利用されている。出力データが与えられない「教師なし学習」はクラスタリングなどに用いられる。「強化学習」では、出力に評価（報酬）を設定し、この評価を最大化するように試行する。このようないくつかの学習方法のそれぞれに対して、様々なアルゴリズムが提案されている。

例えば教師あり学習については、ロジスティック回帰（Logistic Regression）、k近傍法（k-Nearest Neighbors）、決定木（Decision Tree）、サポートベクターマシン（Support Vector

Machine)、ランダムフォレスト（Random Forest)、単純パーセプトロン（Perceptron)、多層パーセプトロン（Multilayer perceptron）などがあり、これらは利用目的やデータの種類によって使い分けされている。

ここで挙げた種々の機械学習の手法を用いて、自分でプログラムを実装していくのは煩雑であり、また独自実装したプログラムによるデータ処理については信頼性の面でも大きな問題となる。そのような煩雑さを取り除いて、機械学習プログラムを比較的手軽に利用するクラウドプラットホームとしてGoogle Cloud AI HubやAzure Machine Learning Studioなどがある。一方、非クラウドで様々な手法を統一的に利用可能にしたのがオープンソース機械学習ライブラリscikit-learnである［6]。scikit-learnは軽量スクリプト言語Pythonで記述されており、様々なオペレーティングシステムや計算機プラットホームで利用可能となっている。従って、処理できるデータの一群（データセット）さえ手元にあれば、手近にあるコンピュータを使って機械学習処理を実践していくことが比較的容易になっている。

3. 研究手法

3.1 計算の概要

本稿ではオープンソース機械学習ライブラリscikit-learnを用いた機械学習計算により、昨年までの授業のクラス分けの処理データ(訓練データ)から学習モデルを構築する。今回の訓練データは出力結果（ここでは実際のクラス分けの結果）を含んでいるため、「教師あり学習」で学習モデルを構築する。この教師あり学習モデルを今年度のクラス分けの処理データ（本番データ）に適用する。なお、本番データについても既にクラス分け処理を行った結果を伴っていることから、機械学習での予測と実際のクラス分けの結果を比較して機械学習モデルの妥当性を考察する。

3.2 機械学習の計算機環境とプログラム

本研究においてオープンソース機械学習ライブラリscikit-learnを用いた機械学習処理のための計算機環境を表 1 にまとめる。また表 1 の計算機環境で使用したPython用のライブラリを表 2 にまとめる。 これらのシステム及びライブラリを用いて、機械学習モデルの構築とモデルの妥当性の検証を行った。

機械学習やモデルの妥当性の検証のためのPythonプログラムについては、インターネット上のいくつかのWebサイトの記事を参考にした［7］［8］［9］［10］［11]。

表 1　機械学習用計算機環境

ハードウェア	AMD Athlon 240GE / RAM 32GB / HDD 8TB システム
OS	FreeBSD 12.2-RERELASE（amd64）
処理系	Python 3.7

表 2　Python用ライブラリ

機械学習ライブラリ	scikit-learn 0.22
数値計算ライブラリ	numpy-1.16.6
グラフ描画ライブラリ（NumPyに依存）	matplotlib-2.2.4
データ解析支援ライブラリ	pandas-0.24.2

3.3 訓練データの概要

本稿で使用するクラス分けの訓練データは、2016年度から2019年度の各年度の情報リテラシー科目履修希望者およそ 1000 人についての意識調査と試験結果およびクラス分けの結果である。ただし、2016 年度の試験については希望者のみの受験としたため、意識調査と試験結果の両方が揃っている資料だけを取り扱うこととなり、機械学習の学習結果に影響を与える可能性があることは予め理解しておく必要がある。各年度で意識調査アンケート並びに試験は複数のクラスで個別に実施したが、調査の条件（設問内容、アンケート回答時間 15 分、試験の解答時間 30 分）はどのクラスでも同じである。調査はマークシート方式および同等のWebアンケート方式で、それぞれに設問について、複数の候補から選択して回答もしくは解答を選択する方式である。意識調査アンケートは 11 項目、試験部分は 50 項目である。なお、アンケートの設問内容および試験の設問概要は文献［12］を参照されたい。

クラス分けについては、各年度について、異なる学科や専攻毎に複数授業が開講され、その授業単位毎にクラス分けを行っている。このことから、訓練データには、学科・専攻の種別のデータも付している。

クラス分けの実際の作業は次の①〜③の様に行った。

① クラスの分割数は 3 である。

② 試験の総合得点度数分布を描き、概ねちょうどよい人数配分になるような度数分布の切れ目（閾値）が見つかればそこでクラスを分割する。

③ ちょうどよい切れ目がなければ、分割境界線上の学生に対して、意識調査結果を考慮して経験や意欲を推し量ることでクラスを分割する。

この①〜③の作業の結果が訓練データの出力結果（クラス分けの結果）となっている。

なお、意識調査および試験の実施年のデータがうまくスケーリングできるように調査年とは全

く異なる 2000 年と 2100 年のダミーデータをそれぞれ 10 件程度追加している。このダミーデータでは、クラス分けの結果が明らかになるようなデータ、例えば試験問題は全問正解であるようなデータで構成している。

3.4 訓練データと学習手法の簡易検証

訓練データの妥当性と学習モデルの妥当性について簡便な検証を行う。この簡易検証のため、クラス分け訓練データを表 3 に示す 5 パターンで用意し、それぞれのパターン毎に学習モデルの構築を試みた。今回は教師あり学習の学習モデルとして、次の（a）から（h）に示す代表的な学習モデルを採用した。ただし、簡易検証のため学習モデルの最適化（ハイパーパラメータチューニング）は行わず、初期値のままで学習させている。なお、各学習モデルの概要については文献［13］［14］［15］［16］などが参考になる。

この簡易検証では、表 3 に示した各訓練データセットパターンのそれぞれにおいて、ランダムに選択された 70% を学習データとして用い、残りの 30% を検証データとして使用した場合について 8 種の学習モデルの検証結果を算出し、さらにこれを 10 回繰り返して算術平均をとった。この訓練データのパターンごとの結果をそれぞれ表 4 から表 8 に示す。表 4 から表 8 の（a）から（h）は前述の学習モデルを示しており、それぞれのモデルで検証した正解率の 10 回の算術平均を平均値として記載している。また、正解率の 10 回の標準偏差、正解率の最大値および最小値、最大値と最小値の差、正解率の平均値が高い方からの順位および正解率の標準偏差の低い方からの順位を記している。分かり易さのため、順位 1 位については太枠で囲んでおり、対応する数値は太字で記している。

表 4 から表 8 の結果から次の（ア）から（オ）のことが言える。

（ア）パターン 1 からパターン 5 のデータセットでは、データセット毎に正解率に差が生じている。またもっともよい成績の学習モデルがデータセットによって異なる。パターン 1 では（e）サポートベクタマシン（rbf カーネル法）、パターン 2 では（f）ランダムフォレスト、パターン 3 からパターン 5 では（c）の決定木がそれぞれ最も成績が良かった。

（イ）訓練データの各パターンに 8 種の学習モデルの成績が依存していることから、8 種の学習モデルの正解率の平均をとったものを表 9 に示す。この表 9 から、パターン 3 が訓練データとしては最も成績がよく、この時の最も成績のよい学習モデルは（c）の決定木であることが分かる。

（ウ）パターン 3 からパターン 5 の結果がパターン 1 およびパターン 2 に比べて似通っているのは、データセットに総合得点が含まれているからであると考える。これは、前述の 3.3 節で述べたように、訓練データとしている過去のクラス分けの結果については総合得点を第一基準としていることから、総合得点を含むパターンのデータセットの成績が良い。

（エ）教師データが総合得点をクラス分けの基準として分割することから、学習モデルとして分岐分類の（c）決定木が最も成績が良くなったと推定される。

（オ）パターン 1 からパターン 5 で平均値が 1 位の学習モデルの標準偏差の順位は必ずしも 1 位ではないが、それぞれのパターンの標準偏差の 1 位と同程度であり、それほど大きなばらつきではないと判断してよい。

以上から、パターン 1 からパターン 3 の訓練データが比較検討を行うための有望な訓練データであり、（c）の決定木、（e）サポートベクタマシン（rbfカーネル法）、（f）ランダムフォレストの 3 種が特に有望な学習モデルである。従って、パターン 1 からパターン 3 の訓練データと上記 3 種の学習モデルの組み合わせの内のいずれかがクラス分け自動化に最適な組み合わせと成り得ると考えられる。

表 3　5 パターンの訓練データ

パターン 1	前節 3.3 で説明した基本データセット
パターン 2	パターン 1 で試験項目 50 項目を特徴的な 12 項目に絞ったデータセット
パターン 3	パターン 1 で試験項目 50 項目を総合得点に置き換えたデータセット
パターン 4	パターン 1 に総合得点を加えたデータセット
パターン 5	パターン 2 に総合得点を加えたデータセット

本稿で使用した学習モデルのリストと略号

（a）ロジスティック回帰（Logistic Regression）

（b）k 近傍法（k-Nearest Neighbors）

（c）決定木（Decision Tree）

（d）サポートベクターマシン（線形）（Support Vector Machine linear）

（e）サポートベクターマシン（rbfカーネル法）（Support Vector Machine kernel method）

（f）ランダムフォレスト（Random Forest）

（g）単純パーセプトロン（Perceptron）

（h）多層パーセプトロン（Multilayer perceptron）

表４　パターン１

	平均値	標準偏差	最大	最小	最大－最小	平均値順位（降順）	標準偏差順位（昇順）
(a)	0.7908012	0.0234174	0.8219580	0.7448070	0.0771510	3	4
(b)	0.6451040	0.0253802	0.6735910	0.6023740	0.0712170	6	7
(c)	0.6178041	0.0237718	0.6439170	0.5845700	0.0593470	8	5
(d)	0.7970326	0.0226074	0.8308610	0.7507420	0.0801190	2	3
(e)	0.8080118	0.0127476	0.8219580	0.7863500	0.0356080	1	1
(f)	0.7270030	0.0208419	0.7507420	0.7002970	0.0504450	5	2
(g)	0.6290802	0.0785712	0.7091990	0.4362020	0.2729970	7	8
(h)	0.7347180	0.0241555	0.7715130	0.6943620	0.0771510	4	6

表５　パターン２

	平均値	標準偏差	最大	最小	最大－最小	平均値順位（降順）	標準偏差順位（昇順）
(a)	0.6130565	0.0229082	0.6587540	0.5845700	0.0741840	2	4
(b)	0.5578635	0.0203190	0.5875370	0.5252230	0.0623140	5	3
(c)	0.5356084	0.0194959	0.5637980	0.4955490	0.0682490	7	2
(d)	0.6097922	0.0244395	0.6498520	0.5756680	0.0741840	3	6
(e)	0.6029673	0.0237718	0.6261130	0.5489610	0.0771520	4	5
(f)	0.6139465	0.0148336	0.6320470	0.5875370	0.0445100	1	1
(g)	0.4934719	0.0795617	0.5608310	0.3353120	0.2255190	8	8
(h)	0.5528191	0.0262833	0.5964390	0.5192880	0.0771510	6	7

表６　パターン３

	平均値	標準偏差	最大	最小	最大－最小	平均値順位（降順）	標準偏差順位（昇順）
(a)	0.7545993	0.0196360	0.7804150	0.7210680	0.0593470	6	3
(b)	0.6317508	0.0211889	0.6528190	0.5786350	0.0741840	8	5
(c)	0.9317508	0.0135620	0.9495550	0.9080120	0.0415430	1	1
(d)	0.7896142	0.0199030	0.8130560	0.7566770	0.0563790	4	4
(e)	0.7584569	0.0238704	0.7863500	0.7091990	0.0771510	5	6
(f)	0.9178041	0.0163159	0.9376850	0.8902080	0.0474770	2	2
(g)	0.6540061	0.0856968	0.7566770	0.4925820	0.2640950	7	8
(h)	0.8228487	0.0271981	0.8605340	0.7774480	0.0830860	3	7

表７　パターン４

	平均値	標準偏差	最大	最小	最大－最小	平均値順位（降順）	標準偏差順位（昇順）
(a)	0.7807122	0.0219598	0.8130560	0.7477740	0.0652820	5	6
(b)	0.6590505	0.0200014	0.6913950	0.6231450	0.0682500	7	4
(c)	**0.9056380**	0.0164204	0.9287830	0.8813060	0.0474770	1	3
(d)	0.7913946	0.0218079	0.8189910	0.7626110	0.0563800	3	5
(e)	0.7910980	**0.0124488**	0.8100890	0.7744810	0.0356080	4	1
(f)	0.8646882	0.0137199	0.8931750	0.8427300	0.0504450	2	2
(g)	0.6314540	0.0509719	0.6913950	0.5637980	0.1275970	8	8
(h)	0.7261128	0.0240684	0.7655790	0.6913950	0.0741840	6	7

表８　パターン５

	平均値	標準偏差	最大	最小	最大－最小	平均値順位（降順）	標準偏差順位（昇順）
(a)	0.7311571	0.0209451	0.7596440	0.6884270	0.0712170	5	4
(b)	0.5881305	0.0267173	0.6379820	0.5489610	0.0890210	8	7
(c)	**0.9166173**	0.0128546	0.9376850	0.9020770	0.0356080	1	2
(d)	0.7569734	0.0212809	0.7952520	0.7181010	0.0771510	3	5
(e)	0.6489614	0.0193345	0.6884270	0.6172110	0.0712160	6	3
(f)	0.9020773	**0.0111907**	0.9228490	0.8872400	0.0356090	2	1
(g)	0.6486647	0.0472131	0.7270030	0.5667660	0.1602370	7	8
(h)	0.7403559	0.0250328	0.7715130	0.6973290	0.0741840	4	6

表９　パターンごとの平均

	平均値	標準偏差	最大	最小	最大－最小	平均値順位
パターン１	0.8080118	0.0786173	0.8080118	0.6178041	0.1902077	4
パターン２	0.6139465	0.0445446	0.6139465	0.4934719	0.1204746	5
パターン３	**0.9317508**	0.1087947	0.9317508	0.6317508	0.3000000	1
パターン４	0.9056380	0.0939604	0.9056380	0.6314540	0.2741840	3
パターン５	0.9166173	0.1180210	0.9166173	0.5881305	0.3284868	2

3.5 学習モデルの最適化

前節 3.4 ではいくつかの訓練データや機械学習モデルの中から、有望な訓練データと学習モデルを見出した。この訓練データと学習モデルを用いて、学習モデルのハイパーパラメータの最適化を行い、最適化していない場合との比較を行う。ここでは、パターン１からパターン３の訓練データに対して、(c) 決定木、(e) サポートベクタマシン（rbfカーネル法)、(f) ランダムフォ

レストをそれぞれ適用して最適化を行った。最適化の手法としては、グリッドサーチ（Grid Search）によるクロスバリデーション（Cross validation; 交差検証）で行った。グリッドサーチは指定した各ハイパーパラメータの組み合わせを総当たりで検証する方法であり、検証するハイパーパラメータの範囲を任意に指定することができる。今回検証する 3 パターンの訓練データに対して同じハイパーパラメータのグリッド（格子点）で検証可能となるため、比較可能な検証結果を得ることができる。ただし、パラメータのグリッド設定数によって計算時間が長大になる傾向がある。因みに今回の計算では、3 種類のデータパターンでグリッドサーチを同時に行ったが、計算終了までに 4 〜 5 日ほどかかっている。グリッドサーチではハイパーパラメータのグリッド部分しか検証しないので、得られる最適化の結果はあくまで検証したグリッドの中での最善のパラメータにすぎないことに注意が必要である。ハイパーパラメータの最適化のプログラムについては、インターネット上の Web サイトの記事を参考にした［17］［18］［19］［20］［21］［22］［23］［24］。

学習モデルごとに最適化可能なハイパーパラメータは異なっているため、学習モデルごとのハイパーパラメータのグリッド設定値を表 10 に示す。なお、クロスバリデーションの分割数は 5 として検証を行った。

表 10　ハイパーパラメータのグリッド設定値

使用した学習モデル	ハイパーパラメータ名	ハイパーパラメータのグリッド設定値
(c) 決定木	criterion	'gini', 'entropy'
	splitter	'best', 'random'
	max_depth	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
	min_samples_split	2, 3, 4, 5, 6, 7, 8, 9, 10
	min_samples_leaf	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
	random_state	0, 1, 2,…, 100
(e) サポートベクタマシン（rbf カーネル法）	C	10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}, 10^{6}
	gamma	10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}, 10^{6}
(f) ランダムフォレスト	criterion	'gini', 'entropy'
	n_estimators	3, 10, 100, 1000, 10000
	max_depth	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
	min_samples_split	2, 3, 4, 5, 6, 7, 8, 9, 10
	min_samples_leaf	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
	random_state	0
	bootstrap	True, False

3.6 学習モデル最適化の効果

前節 3.5 で述べた設定でクロスバリデーションによる学習モデルの最適化を行った。表 11 に最適化前と最適化後の正答率スコアを示す。この表の最適化後訓練スコアと最適化前スコアとの比がスコア向上率（対訓練）として示されており、パターン 1 の（c）決定木とパターン 2 の（e）のサポートベクタマシン以外は 1 を超えている。典型的には正答率が対訓練では 1 割程度向上していることから、全体としてグリッドサーチによるモデル最適化は正答率の向上に寄与していると考えてよい。このことを踏まえ、訓練データのパターン 1 からパターン 3 について最適化の結果の特徴をそれぞれ（i）から（iii）に述べる。

（i）訓練データ　パターン 1 の最適化

訓練データパターン 1 のグリッドサーチの結果を表 12 に示す。この訓練データパターンでは（e）サポートベクタマシンが訓練データでの正答率が最も高く、テストデータにおける正答率ではこのパターン内では最も高くなっている。（f）ランダムフォレストは訓練スコア、テストスコアともにサポートベクタマシンと同程度であり、訓練スコアとテストスコアの差異も 2 割程度向上と同じような傾向を示している。（c）決定木については、他の二つと比べて相対的に正答率が低いが、ハイパーパラメータチューニングによる正答率の向上は、他の二つと同程度である。表中の順位は他のデータパターンも含めた順位になっているが、このデータパーンだけで判断すると、訓練スコアとテストスコアの正答率の高さから（e）サポートベクタマシンが最も良い結果である。

（ii）訓練データ　パターン 2 の最適化

訓練データパターン 2 のグリッドサーチの結果を表 13 に示す。この訓練データパターンでは（f）ランダムフォレストが訓練データおよびテストデータでの正答率が最も高い。しかし、データパターン 1 やデータパターン 3 の結果と比較すると、この正答率もかなり低い。（c）決定木と（e）サポートベクタマシンはほぼ同じスコアであり、またパラメータチューニングによる正答率の向上も 1 割程度と同じような傾向を示している。他のデータパターンと比較すると、このデータパターンによる学習では正答率が 0.7 程度以下であり、自動クラス分けは十分機能しないと予想される。

（iii）訓練データ　パターン 3 の最適化

訓練データパターン 3 のグリッドサーチの結果を表 14 に示す。この訓練データパターンでは（f）ランダムフォレストが訓練データでの正答率が最も高く、テストデータにおける正答率では（c）決定木が最も高くなっている。（e）サポートベクタマシンは訓練スコア、テストスコアともに若干正答率が低いが概ね同程度であると考えられる。訓練による正答率

の向上は、どの学習モデルでも 5%～ 6% の向上幅にとどまっている。

以上の（i）から（iii）から今回の学習モデルの最適化で最も良好であるのは訓練データパターン 1 の（e）サポートベクタマシンであることが分かる。しかし、テストスコアでは訓練データパターン 3 の（c）決定木が最も良好である。

表 11　学習モデル最適化の効果

※表中の太字は最良スコア、斜体はスコア向上が 1 未満のスコアを示す

データセット	学習モデル	最適化後訓練スコア	最適化後テストスコア	最適化前スコア	スコア向上率（対訓練）	スコア向上率（対テスト）
パターン 1	(c) 決定木	0.7468193	0.6023739	0.6178041	1.2088287	*0.9750241*
	(e) サポートベクタマシン	**1.0000000**	0.8130564	0.8080118	1.2376057	1.0062432
	(f) ランダムフォレスト	0.9923664	0.7388724	0.7270030	1.3650101	1.0163265
パターン 2	(c) 決定木	0.6513995	0.5756677	0.5356084	1.2161861	1.0747921
	(e) サポートベクタマシン	0.6526718	0.5786350	0.6029673	1.0824331	*0.9596458*
	(f) ランダムフォレスト	0.7061069	0.6142433	0.6139465	1.1501114	1.0004835
パターン 3	(c) 決定木	0.9872774	**0.9376855**	0.9317508	1.0595938	1.0063694
	(e) サポートベクタマシン	0.8689567	0.8189911	0.7584569	1.1456903	1.0798123
	(f) ランダムフォレスト	0.9923664	0.9287834	0.9178041	1.0812399	1.0119626

表 12　訓練データ　パターン 1 のハイパーパラメータチューニング結果

学習モデル	訓練スコア		テストスコア		スコア差分		スコア差分相対値	
	スコア	順位	スコア	順位	差分	順位	差分相対値	順位
(c) 決定木	0.7468193	6	0.6023739	7	0.1444455	3	0.1934142	2
(e) サポートベクタマシン	**1.0000000**	1	0.8130564	4	0.1869436	2	0.1869436	3
(f) ランダムフォレスト	0.9923664	2	0.7388724	5	**0.2534940**	1	**0.2554440**	1

表 13　訓練データ　パターン 2 のハイパーパラメータチューニング結果

学習モデル	訓練スコア		テストスコア		スコア差分		スコア差分相対値	
	スコア	順位	スコア	順位	差分	順位	差分相対値	順位
(c) 決定木	*0.6513995*	9	*0.5756677*	9	0.0757318	5	0.1162602	5
(e) サポートベクタマシン	0.6526718	8	0.5786350	8	0.0740367	6	0.1134364	6
(f) ランダムフォレスト	0.7061069	7	0.6142433	6	0.0918635	4	0.1300986	4

表 14　訓練データ　パターン 3 のハイパーパラメータチューニング結果

学習モデル	訓練スコア		テストスコア		スコア差分		スコア差分相対値	
	スコア	順位	スコア	順位	差分	順位	差分相対値	順位
(c) 決定木	0.9872774	4	**0.9376855**	1	*0.0495919*	9	*0.0502310*	9
(e) サポートベクタマシン	0.8689567	5	0.8189911	3	0.0499656	8	0.0575007	8
(f) ランダムフォレスト	0.9923664	2	0.9287834	2	0.0635830	7	0.0640721	7

3.7 クラス分けへ適用への可否

前節 3.6 の結果から、最も成績の良いハイパーパラメータを使い 2016 年から 2019 年までのデータの全てを訓練データとして用い、2020 年度のクラス分けテストの結果を予測した。この予測値に対して手作業によるクラス分けの結果と照合した正答率を訓練データパターンごとに表 15 に示す。

前節 3.6 の結果からは訓練データ　パターン 3 の（c）決定木のテストスコアが最もよい正答率であり、次点が（f）のランダムフォレストであったことから、データパターン 3 の決定木もしくはランダムフォレストの正答率が高いと予測された。しかし、実際には総じてスコアの低かったデータパターン 2 が新規データに対しては予測の正答率が高いことが分かった。このような訓練データと本番データでの正答率の逆転現象が生じた理由については次のように考えられる。

3.3 節の訓練データの概要で述べたように、クラス分けは原則的に各クラスの総合点数の分布グラフにより閾値の点数を決定して分割する。この際、クラス毎に閾値が異なる。実施年度や専攻ごとに閾値が異なるので、データパターン 3 の様に合計得点だけで示されたデータを学習しても、本番データにおける同じ専攻全体の得点分布が学習データと乖離していればクラス分けの予測精度は向上しないと考えられる。これに対して、データパターン 1 は全ての回答項目をデータとして使用しているが、この回答項目のそれぞれの正解状況が合計得点として算出される。またデータパターン 2 で取り上げた特定の項目は合計得点の多寡に大きな影響を与えていると考えられる回答項目だけを抜粋しており、データパターン 1 よりも合計得点の多寡に対する応答がより敏感であると考えられる。このようなデータパターンの性質から、本番データではデータパターン 2 やデータパターン 1 の成績が良くなり、データパターン 3 の成績が大幅に下落する原因となっていると考えられる。

データパターン 2 の（f）ランダムフォレストの結果を実際にクラス分けに用いた場合、テストスコアの正答率が 0.8 程度であり、受講者総数が 200 人程度であることを考えると 160 人程度は正しくクラス分けされるが、40 人程度が誤判定されてしまうことになる。従って、実用を考えた場合には正答率は少なくとも 0.95 以上は必要と考えられることから、今回の最良学習モデルとデータパターンの組み合わせの場合でも、学習モデルの選択やハイパーパラメータチューニングなど改善の余地がある。

改善の方法としては、グリッドサーチのグリッドをさらにきめ細かく設定してハイパーパラメータチューニングを実行してより効果的なハイパーパラメータを求めることに加えて、ハイパーパメータの最適化方法をより高度な手法のベイズ最適化などを用いることが考えられる。また、機械学習の手法として深層学習と呼ばれる手法を適用することで、より良好な予測精度を持つ学習モデルが得られる可能性がある。

表 15　学習結果の 2020 年度本番データへの適用結果

学習モデル	本番データに対する予スコア		
	データパターン 1	データパターン 2	データパターン 3
(c) 決定木	0.6063830	0.6223400	0.3404260
(e) サポートベクタマシン	0.6808510	0.7978720	0.3510640
(f) ランダムフォレスト	0.7659570	**0.8031910**	0.3563830

4. まとめ

本稿では、scikit-learn を用いた機械学習の応用として、アンケートと試験結果からクラス分けの自動化を検討した。本研究では試行的に 8 種類の機械学習モデルと 5 種類のデータセットのパターンを組み合わせて比較検討した。今回検討した学習モデルとデータセットについて、学習時のテストでの最良の組み合わせが本番データに対しては最も悪い成績となり、学習時テストではあまり良い正答率ではなかった組み合わせが本番データに対しては最良の予測結果となった。本番データに対する本稿での最良の組み合わせは、アンケート項目から得点に強い影響を与えると考えられる項目だけを抜き出した「データパターン 2」に対して、学習モデル「ランダムフォレスト」を適用してグリッド―サーチによる最適化を行った場合であった。この最良の組み合わせでは正答率が 0.8 程度であることから、機械学習をクラス分けに応用することは有望である一方、クラス分けの自動化を実用的に行う上では未だ改善の余地があることが分かった。これを改善するためには、グリッド数を増やしてハイパーパラメータチューニングを実行し、さらに良いハイパーパラメータを発見するということ以外に、より高度なハイパーパラメータの最適化手法を使用することが考えられる。あるいは深層学習などのより高度な機械学習手法を用いることでより高精度な予測が可能なモデルを構築できる可能性がある。

今後、より大規模なグリッドサーチや深層学習に耐えられるように計算機リソースを見直したうえで先に述べた改善策を検証してくことが求められる。

5. 謝辞

本稿の執筆にあたり、プログラミング教育研究会において多くの示唆に富む議論や助言を頂戴いたしました梅光学院大学　子ども学部　横山　修　准教授、 梅光学院大学　子ども学部　4 年　福永　光一朗　氏に厚く御礼申し上げます。

参照文献目録

1. 自分で構築するか，APIで機能を使うか．木村 優志昌平，井上 祐寛，小川 雄太郎泰．東京都新宿区：技術評論社，2018年3月，Software Design 2018年4月号.
2. Some studies in machine learning using the game of checkers. SamuelL.A. 3, NEW ORCHARD ROAD, ARMONK, USA, NY, 10504 : IBM, 1959 年 7 月，IBM Journal of Research and Development，第3巻，ページ: 210-229.
3. Some Studies in Machine Learning Using the Game of Checkers. II—Recent Progress. SamuelL.A. 6, NEW ORCHARD ROAD, ARMONK, USA, NY, 10504 : IBM, 1967 年 11 月，IBM Journal of Research and Development，第11巻，ページ: 601-617.
4. 変わりゆく機械学習と変わらない機械学習．敏弘神嶌．1, 2019年1月, 日本物理学会誌, 第74巻, ページ: 5-13.
5. 総務省．総務省 ICTスキル総合習得プログラム．3-5 人工知能と機械学習．（オンライン）（引用日: 2020年9月16日.）https://www.soumu.go.jp/ict_skill/pdf/ict_skill_3_5.pdf.
6. Scikit-learn: Machine Learning in Python. al.etPedregosa．2011年，Journal of Machine Learning Research，第12巻，ページ: 2825--2830.
7. 小川 雄太郎．Yutaro Ogawa scikit-learn_tutorial_SoftwareDesign ロジスティック回帰実装（SoftwareDesign 2018年4月号　特集「機械学習の始め方」第4章の掲載コード）．Github.（オンライン）2018年3月1日．（引用日: 2020年9月11日.）https://github.com/YutaroOgawa/scikit-learn_tutorial_SoftwareDesign/blob/master/program/logistic_regression_SD1804.ipynb.
8. Kenta Sasaki. scikit-learn（sklearn）の使い方．Qiita.（オンライン）2017年11月8日．（引用日: 2020年9月11日.）https://qiita.com/kenta1984/items/c2f3b2609071717dcf71.
9. kurita．機械学習やるならGoogle Colabが素晴らしかった話（Python実行環境）．note.（オンライン）2020年3月30日．（引用日: 2020年9月11日.）https://note.com/mc_kurita/n/n2a7c8682d965.
10. LOCALAB. Python scikit-learnで機械学習モデルを保存＆ロードする．LOCALAB.（オンライン）2017年8月7日．（引用日: 2020年9月11日.）https://localab.jp/blog/save-and-load-machine-learning-models-in-python-with-scikit-learn/.
11. でんだろ．Scikit-Learnの識別器をループで一気に適用させる．AI人工知能テクノロジー.（オンライン）2019年1月27日．（引用日: 2020年9月11日.）https://newtechnologylifestyle.net/scikit-learn-ikkatu/.
12. Society 5.0時代に向けた文系大学の教育内容の検討．高瀬　剛．53，山口県下関市：梅光学院大学，2020年3月31日，梅光学院大学 論集，ページ: 1-27.
13. Octoparse. 機械学習に知っておくべき10のアルゴリズム．Octoparse.（オンライン）2020年5月8日.（引用日: 2020年9月11日.）https://www.octoparse.jp/blog/10-machine-learning-algorithms-you-should-know/#div4.
14. AI drops．機械学習の代表的なアルゴリズム19選．AI drops.（オンライン）2019年8月29日．（引用日: 2020年9月30日.）https://www.bigdata-navi.com/aidrops/1163/.
15. SAS Japan翻訳・編集（原文Hui Li著）．機械学習アルゴリズム選択ガイド．SAS Blogs.（オンライン）2017年11月21日．（引用日: 2020年9月30日.）https://blogs.sas.com/content/sasjapan/2017/11/21/machine-learning-algorithm-use/.

16. TRaiNZ. 機械学習アルゴリズムの分類別代表例と効果的な学習方法を解説. TRaiNZ.（オンライン）2020 年 7 月 21 日.（引用日: 2020 年 9 月 30 日.）https://trainz.jp/media/learningtoai/tecnical-term/1700/.
17. tomov3. scikit-learn を用いた交差検証（Cross-validation）とハイパーパラメータのチューニング（grid search）. Qiita.（オンライン）2019 年 7 月 29 日.（引用日: 2020 年 9 月 11 日.）https://qiita.com/tomov3/items/039d4271ed30490edf7b.
18. ハイパーパラメータとは？チューニングの手法を徹底解説（XGBoost 編）. codexa.（オンライン）2020 年 3 月 31 日.（引用日: 2020 年 9 月 11 日.）https://www.codexa.net/hyperparameter-tuning-python/.
19. momijiame. Python: scikit-learn のハイパーパラメータを GridSearchCV で最適化する. CUBE SUGAR CONTAINER.（オンライン）2017 年 9 月 5 日.（引用日: 2020 年 9 月 11 日.）https://blog.amedama.jp/entry/2017/09/05/221037.
20. FujiedaTaro. ロジスティック回帰（分類）とハイパーパラメータのチューニング. Qiita.（オンライン）2019 年 11 月 14 日.（引用日: 2020 年 9 月 11 日.）https://qiita.com/FujiedaTaro/items/5784eda386146f1fd6e7.
21. —. 非線形 SVC（分類）のハイパーパラメータとチューニング. Qiita.（オンライン）2019 年 11 月 14 日.（引用日: 2020 年 9 月 11 日.）https://qiita.com/FujiedaTaro/items/e5583f8767173e6a6f9a.
22. wat@watlablog. Python のグリッドサーチで決定木のハイパーパラメータを調整！ WATLAB.（オンライン）2020 年 2 月 20 日.（引用日: 2020 年 9 月 11 日.）https://watlab-blog.com/2020/02/20/grid-search-decisiontree/.
23. Scikit-learn によるロジスティック回帰. データ科学便覧.（オンライン）2017 年.（引用日: 2020 年 9 月 11 日.）https://data-science.gr.jp/implementation/iml_sklearn_logistic_regression.html.
24. でんだろ. SckitLearn でグリットサーチを行いパラメータを最適化する. AI 人工知能テクノロジー.（オンライン）2018 年 7 月 7 日.（引用日: 2020 年 9 月 11 日.）https://newtechnologylifestyle.net/scikitlearn%E3%81%A7%E3%82%B0%E3%83%AA%E3%83%83%E3%83%88%E3%82%B5%E3%83%BC%E3%83%81%E3%82%92%E8%A1%8C%E3%81%84%E3%83%91%E3%83%A9%E3%83%A1%E3%83%BC%E3%82%BF%E3%82%92%E6%9C%80%E9%81%A9%E5%8C%96%E3%81%99/.
25. 機械学習アルゴリズムの分類別代表例と効果的な学習方法を解説. TRA.（オンライン）https://trainz.jp/media/learningtoai/tecnical-term/1700/.