

Scratchにおけるコンピューティショナル・シンキングスキル に基づく学習者の習熟度到達予測

安東 亮汰^{1,a)} 伊原 彰紀^{1,b)}

概要: 本研究は、Scratch を利用するプログラミング学習者が作品制作に使用するコンピューティショナル・シンキング (CT) スキルの概念を明らかにし、学習者が新たに制作する作品が特定の習熟度に到るか否かを予測するモデル (習熟度到達予測モデル) を提案する。Scratch において作品を 20 回以上制作した学習者 6,323 人を対象に評価実験を行った結果、習熟度到達予測モデルは適合率 0.93 以上、再現率 0.83 以上の精度で予測することができ、学習者が特定の習熟度の作品を制作するためには、CT スキルの概念「論理」や「同期」の習熟が重要であることを明らかにした。

1. はじめに

初等教育からのプログラミング教育が進められており、プログラミング教材の一つとしてビジュアルプログラミングの学習環境 Scratch^{*1}がある。Scratch は、プログラムの命令処理をブロックで視覚的に表現することで、学習者の直感的な作品制作を実現している。Scratch は、テキストによるプログラミング教育と同様に、学習者がプログラムの実装を通してコンピューティショナル・シンキングスキル (CT スキル: 問題を抽象化し分析することで、効率的な問題解決を可能にする考え方) [1] を習得するために広く利用されている。また、Scratch において、学習者の CT スキルを評価するためのツールには Dr.Scratch [2] がある。Dr.Scratch は、作品中のプログラムに使用されるブロックを解析し、作品の機能実装に使用される CT スキル 7 概念をそれぞれ 0 点から 3 点で算出し、合計点数 0 点から 21 点までの 22 段階で作品を評価する。特に、合計点数により CT スキルの習熟度を 3 つに分け、0 点から 7 点を Basic, 8 点から 14 点を Developing, 15 点から 21 点を Master としている [3]。

数学や英語のように明確なカリキュラムを有する学習はフォーマル・ラーニングに分類される一方で、学習者が自由に作品制作を進めながらプログラミングを学習する Scratch はインフォーマル・ラーニングに分類される。インフォーマル・ラーニングである Scratch は、学習者自らが

進んで学習を行う環境を提供しているが、明確な学習カリキュラムを提示していない。そのため、高い CT スキルを必要とする作品の制作に向けて、学習者が習熟すべき CT スキルの概念を把握することは困難である。インフォーマル・ラーニングにおける学習習熟度の把握を目的とした Yang らの研究では、Scratch における学習者は多数の作品制作を経て、使用ブロックの種類が増加することを明らかにしている [4]。この結果から、学習者は複数回の作品制作を通して、高い CT スキルを必要とする作品制作が可能になることが示唆される。しかし、学習者が高い CT スキルを必要とする作品を制作するまでに、過去に制作した作品で使用した CT スキルの概念は明らかではない。

本研究では、Scratch において特定の習熟度に到達する作品を制作するまでに、学習者が過去に制作した作品で使用した概念を明らかにする。さらに、学習者が過去に使用した CT スキルの概念に基づき、学習者が新たに制作する作品が特定の習熟度に到るか否かを予測するモデルを構築する。事前分析として、Scratch において 20 件以上の作品を制作した学習者 6,323 人の習熟過程を調査し、Dr.Scratch が算出する CT スキルの点数は、多数の作品制作を実施することで高くなるのか否かを明らかにする。

事前分析の結果から、学習者が過去に獲得した CT スキルの概念の点数に基づき、新たに制作する作品が習熟度 Developing 以上 (合計点数 8 点以上)、または習熟度 Master (15 点以上の作品) に到達するか否かを予測するモデルを構築する。本研究により、Scratch において学習者が高い CT スキルを要する作品を制作するまでに習熟した CT スキルの概念を明らかにすることで、プログラミング学習者が有する CT スキルの把握、および Scratch におい

¹ 和歌山大学
Wakayama University, Wakayama 640-8510, Japan
^{a)} s216009@wakayama-u.ac.jp
^{b)} ihara@wakayama-u.ac.jp
^{*1} Scratch: <https://scratch.mit.edu/>

表 1 Dr.Scratch による CT スキル 7 概念の評価方法 [2]

CT スキルの概念	0 点	1 点	2 点	3 点
抽象化	-	2 つ以上のスクリプトを使用	定義ブロックを使用	クローンブロックを使用
並列	-	緑の旗ブロックを 2 個以上使用	オブジェクトへのクリック動作により 2 つ以上のスクリプトを同時に 実行する機能を実装	背景が切り替わったとき, メッセージを受け取ったとき等, 様々なイベント動作により 2 つ以上の スクリプトを同時に実行する機能を実装
論理	-	If ブロックを使用	If else ブロックを使用	論理演算ブロックを使用
同期	-	待機ブロックを使用	メッセージを受信すると プログラムを停止する機能を実装	指定条件を満たすまで プログラムを停止する処理を実装
フロー制御	-	2 個以上の処理ブロックを連結して使用	指定回数/回数無制限の繰り返しブロックを使用	指定条件までの繰り返しブロックを使用
ユーザ対話性	-	緑の旗ブロックを使用	ユーザによる入力を伴うブロックを使用	マイクやビデオなどの インタラクションを伴うブロックを使用
データ表現	-	オブジェクトの大きさや位置等の プロパティを編集	変数ブロックを使用	リスト変数ブロックの使用

て公開される作品の中から学習者の CT スキルに合わせた作品の推薦に貢献できると考える。

続く 2 章では, Scratch と関連研究について述べ, 本研究の立ち位置を説明する. 3 章で予測モデルで使用する説明変数の決定に向けた事前分析について述べる. 4 章では習熟度到達予測モデルの構築方法を述べ, 5 章で Scratch に記録された作品に基づく予測モデルの評価結果を述べる. 6 章で結果の考察を述べ, 7 章で本研究をまとめる.

2. Scratch を用いたプログラミング学習

2.1 Scratch

Scratch^{*2}は, MIT メディアラボが開発しているビジュアルプログラミング言語の 1 つであり, ブロックのような視覚的なオブジェクトを組み合わせることで直感的なプログラミングを実現している. テキストベースのプログラミング言語を使用した学習と比較して, Scratch は文字入力の間違いによるエラーが発生しないため学習難易度が低く, プログラミング初学者の学習ツールとして利用されることが多い. 従来研究では, Scratch を用いた学習がプログラミング学習に有効であり, テキストベースのプログラミングへの移行を容易にすることを実証実験により明らかにしている [5]. 学習者は制作した作品を Scratch サービス上に公開することができる. また, Scratch では作品を複製し, 再利用する機能「リミックス」が提供されており, 学習者はリミックスにより他の作品を複製し, 機能の追加や削除を行うことで新たな別の作品として制作することが可能である.

2.2 作品解析ツール: Dr.Scratch

Scratch を使用している学習者は, 作品制作を通して CT スキルを身に付ける. Moreno らは学習者が制作した Scratch の作品に必要な CT スキルを計測する作品評価ツール Dr.Scratch を開発している [2]^{*3}. Dr.Scratch は, 作品内で使用されているブロックやプログラムの構造を解析し, CT スキルの 7 概念 (抽象化, 並列, 論理, 同期, フ



図 1 Dr.Scratch の評価画面 (作品 <https://scratch.mit.edu/projects/311354215> に適用)

ロー制御, ユーザ対話性, データ表現) を計測する. 表 1 は, 各概念の計測方法を示す. 各概念には作品中で使用するブロックの種類, 数に基づき, 0 点から 3 点の点数が算出され, それら 7 概念の点数から学習者が制作した作品を合計点数 0 点から 21 点までの 22 段階で評価する. 特に, 合計点数により CT スキルの習熟度を 3 つに分け, 0 点から 7 点を Basic, 8 点から 14 点を Developing, 15 点から 21 点を Master としている [3]. 図 1 は, 作品 (a) に対する Dr.Scratch の計測結果 (b) を示す. 図 1 中の (1), (2) に示すように, 作品 (a) は, 抽象化で 1 点, 並列で 3 点, 論理で 3 点, 同期で 3 点, フロー制御で 3 点, ユーザ対話性で 2 点, データ表現で 2 点の合計点数 17 点, 習熟度 Master となる.

2.3 従来研究

世界各地で初等教育段階からのプログラミング教育が本格化しており, CT スキルを育むためにはビジュアルプログラミング言語を用いた教育が有効である [7]. 特に, Scratch は優れたプログラミング学習環境として, 世界中の教育機関で利用されており, Scratch の学習効果に関する研究が進められている [4][8][9][6][10][11].

Scratch で制作された作品の特徴を調査するために, Aivaloglou らは Scratch における学習者 109,960 人が制作した公開作品 250,163 件から, 各作品に含まれるプログラム, および各作品を Dr.Scratch によって評価した結果を収

*2 Scratch: <https://scratch.mit.edu/>

*3 Dr.Scratch: <http://drscratch.org/>

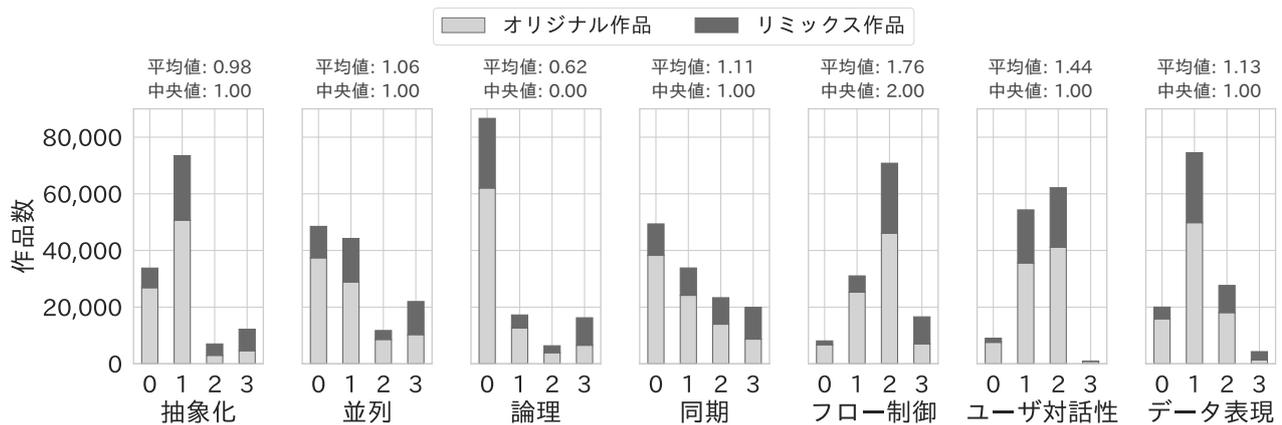


図 2 CT スキル 7 概念の点数分布

集し、公開している [8]. Aivaloglou らは実行可能なプログラムを含む作品 233,491 件を解析した結果、スクリプトの規模は比較的小さく、複雑度も単純であることを明らかにした [12].

Scratch における学習者のプログラミング能力の成長に関する研究として、Yang らは、Scratch で少なくとも 50 回以上の作品制作を行った学習者 3,852 人は、繰り返し作品を制作することで、作品に使用するブロックの種類数が増加することを明らかにした [4]. Troiano らは、8th-grade (13 歳から 14 歳) の生徒 19 人を対象に、それぞれの作品制作過程で Dr.Scratch により作品に使用する概念を調査した結果、作品制作を進めるにつれて並列、論理、同期の点数が高くなる生徒は多いが、抽象化の点数が高くなる生徒は少ないことを明らかにした [10]. また、Dasgupta らは、Scratch において学習者はリミックスを使用して作品を制作することで、新たな CT スキル概念を使用した作品制作が可能になることを明らかにした [6].

多くの従来研究は、学習者が制作した 1 つの作品に使用された概念、または 1 つの作品制作過程において使用される概念の順序を調査し、一部の概念 (抽象化など) を使用した作品制作は困難であること明らかにしている。しかし、学習者が複数の作品制作を重ねることで使用する概念、および使用する概念の順序は明らかにされていない。本研究では、学習者が特定の習熟度に到るまでに作品制作に使用した概念を明らかにし、学習者が次に制作する作品が特定の習熟度以上の評価を得るか否かを予測するモデルを構築する。続く事前分析では、学習者が特定の習熟度の評価を得る作品を制作するまでに、過去に制作した作品の特徴を調査する。

3. 事前分析

本章では、Scratch において制作された作品で実装されたプログラムを収集し、学習者が特定の習熟度に到達するまでに、過去に制作した作品の特徴を調査する。

3.1 Scratch プログラムの収集

Scratch は、2019 年 1 月 3 日のバージョン 3.0 のリリースにおいて、使用可能なブロックを追加するなど大規模なアップデートを実施している。本研究では、学習者間で共通の開発環境で制作された作品を比較するため、バージョン 3.0 をリリースした 2019 年 1 月 3 日から 2020 年 1 月 3 日までに初めて作品制作を行った学習者を分析対象とする。まず、Scratch が提供する API^{*4}を用いて、20 件以上の作品を制作した学習者 7,050 人が制作した作品 141,000 件に関するデータを収集した。その後、収集した作品プログラムの調査、および Dr.Scratch による評価結果の収集を行った結果、Scratch3.0 には存在しないブロック^{*5}を含む作品 (715 件) や Dr.Scratch による評価に失敗した作品 (322 件) が一部存在したため、それらの作品を制作経験に持つ学習者 727 人は分析対象から除外した。最終的に、本研究では学習者 6,323 人が 1 番目から 20 番目までに制作した作品 126,460 件を分析対象とする。

3.2 分析対象作品の特徴分析

図 2 は、収集した作品を対象に Dr.Scratch によって算出した CT スキル 7 概念の点数別作品数の分布を示す。各グラフは、横軸に各概念における点数 [0-3] 点、縦軸に各点数を獲得した作品数を示す。また、グラフ中の灰色で示す箇所はリミックスを使用せずに制作された作品 (オリジナル作品)、黒色で示す箇所はリミックスを使用して制作された作品 (リミックス作品) を示す。図 3 は、合計点数別作品数の分布を示す。横軸は合計点数、縦軸は各点数を獲得した作品数を示す。灰色、黒色で示す箇所は図 2 と同様である。

図 2 から、学習者は、作品制作においてフロー制御の点数を獲得することは容易である一方で、論理の点数を獲

*4 [https://ja.scratch-wiki.info/wiki/Scratch_API_\(2.0\)#GET_.2Fusers.2F.3Cusername.3E](https://ja.scratch-wiki.info/wiki/Scratch_API_(2.0)#GET_.2Fusers.2F.3Cusername.3E)

*5 https://en.scratch-wiki.info/wiki/Experimental_Blocks

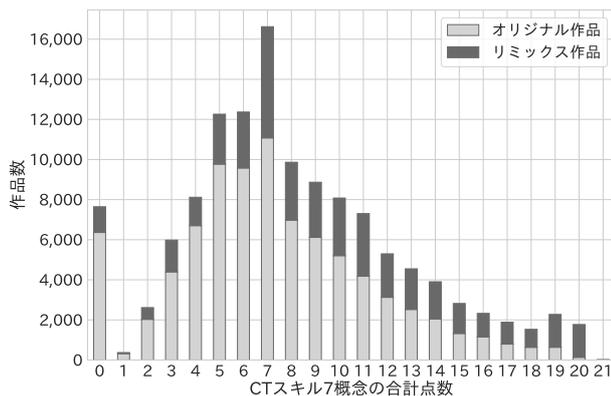


図 3 CT スキル 7 概念の合計点数の分布

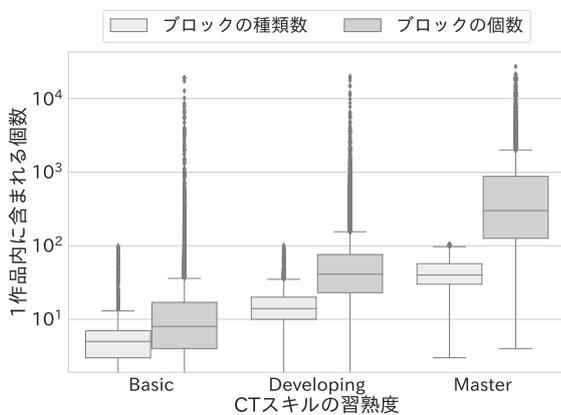


図 4 CT スキルの習熟度別の作品規模の違い

得することは困難であることが考えられる。この結果は、Aivaloglou らが公開する作品と同じ結果を示した [8]。

また、図 3 から、合計点数は 7 点の作品が最頻値であり、8 点の作品以降、合計点数が高くなるにつれて作品数が減少し、Dr.Scratch が分類する CT スキルの習熟度である Developing (合計点数が 8 点から 14 点)、Master (合計点数が 15 点から 21 点) に分類される作品は少ない。各習熟度の作品数は、Basic が 65,976 件、Developing が 47,861 件、Master が 12,623 件である。習熟度による作品規模の違いを図 4 に示す。横軸は 3 種類の習熟度 (Basic, Developing, Master)、縦軸はそれぞれの規模を対数軸で示している。習熟度間における作品規模は統計的有意な差 (マン・ホイットニーの U 検定, 有意水準 5%) を確認できることから、習熟度 Developing や Master に分類される作品は、学習者にとって制作困難であることが示唆される。

3.3 学習者が過去に制作した作品の点数分析

従来研究では、特定の作品 1 件に対する CT スキルの特徴を調査している。本章では、学習者が自身で制作するオリジナル作品 (N 番目に制作した作品) に至るまでに制作した複数の作品 (1 番目から N-1 番目に制作した作品) における CT スキルを調査する。

図 5 は、学習者が N 番目に制作したオリジナル作品において、初めて CT スキルの合計点数を獲得するまでに、過去に制作した作品 (1 番目から N-1 番目に制作した作品) の CT スキルの合計点数別作品数の分布をバイオリン図で示す。横軸は N 番目に制作したオリジナル作品の CT スキルの合計点数、縦軸は 1 番目から N-1 番目に制作した作品の CT スキルの合計点数の分布を示す。各バイオリン図の膨らみが大きい縦軸の点数は、1 番目から N-1 番目に制作した作品の中で最も獲得した回数が多い合計点数を表す。例えば、7 点を獲得した作品は、当該作品を制作するまでに 5 点の作品を制作した回数が最も多い。オリジナル作品 (N 番目に制作した作品) ごとに、左側の灰色の分布は過去に制作したオリジナル作品、右側の黒色の分布は過去に制作したリミックス作品の制作頻度を示す。これらの結果を Dr.Scratch が分類する 3 つの習熟度別に分析結果を述べる。

- **Basic のオリジナル作品を制作するまでの特徴:** 合計点数の中央値が 7 点であり、四分位偏差は 2.0 点である。このことから、学習者は連続して習熟度 Basic、または Developing の作品を制作することが多く、Master の作品を制作することは少ない。
- **Developing のオリジナル作品を制作するまでの特徴:** 合計点数の中央値が 7 点であり、四分位偏差は 2.5 点である。このことから、学習者は Basic のオリジナル作品を制作するまでの特徴と同様に、連続して習熟度 Basic、Developing の作品を制作することが多く、Master の作品を制作することは少ない。
- **Master のオリジナル作品を制作するまでの特徴:** 合計点数の中央値が 10 点であり、四分位偏差は 3.0 点である。このことから、学習者は習熟度 Master の作品を制作するまでに習熟度 Developing の作品を制作することが多い。図 3 から習熟度 Master の作品を制作する学習者は少ないことが分かるように、習熟度 Master の作品を連続して制作することは少ない。

事前分析の結果から、Scratch を使用した学習において、学習者は CT スキルの合計点数が低いほど (特に習熟度 Developing 以下であるほど)、過去に制作した作品の CT スキルの合計点数は同一点数前後であった。したがって、学習者は突如点数の高い作品を制作することはなく、類似する CT スキルを使用した作品を連続で制作していることが示唆される。ただし、オリジナル作品とリミックス作品における分布は異なることから、次に制作する作品に影響しているのがオリジナル作品かリミックス作品であるのかは明らかではない。

続く 4 章では、学習者が過去に制作した作品で使用した概念の特徴量に基づき、学習者にとって到達が困難な目標習熟度 Developing 以上 (8 点以上)、または Master (15 点以上) の評価を得る作品を制作するか否かを予測する学習

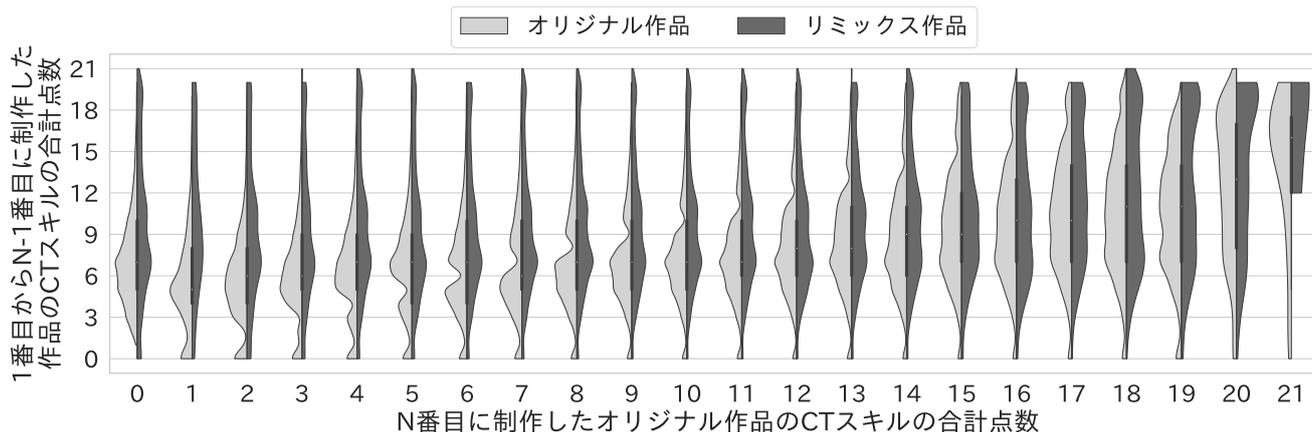


図 5 オリジナル作品 (N 番目に制作した作品) で CT スキルの合計点数を獲得するまでに、過去に制作した作品 (1 番目から N-1 番目に制作した作品) の CT スキルの合計点数別作品数の分布

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	モデル1	モデル2	
学習者-A	B	D	B	B	B	B	B	B	B	D	B	D	B	B	B	B	B	B	B	B	B	負例	負例
学習者-B	B	B	B	M	D	B	B	B	D	B	B	B	B	B	B	B	B	B	B	B	B	正例	負例
学習者-C	B	B	D	D	D	B	D	B	M	M	M	D	M	M	D	D	B	M	M	M	M	正例	正例

B: Basic D: Developing M: Master
 オリジナル作品
 リミックス作品

図 6 2 つのモデルにおける正例・負例クラスの事例

者の習熟度到達予測モデルの構築方法を述べる。また、5章では予測精度の評価結果を述べる。

4. 学習者の習熟度到達予測モデル

4.1 習熟度到達予測モデルの構築

本研究では、特定の目標習熟度に到達する作品を制作する学習者が、過去に制作した作品に使用する CT スキルを明らかにする。具体的には、学習者が Scratch において制作する 1 番目から 20 番目までの作品で、目標習熟度に到達するまでに制作した作品の CT スキルを説明変数とし、目標習熟度に到達する作品を制作する学習者か否かを予測する機械学習モデルを構築する。目標習熟度に到達しない学習者は 1 番目から 20 番目までに制作した作品の CT スキルを説明変数とする。3 章の事前分析の結果に基づき、目標習熟度が異なる 2 種類の予測モデルを構築する。

- (モデル 1) CT スキルの合計点数が 8 点以上 (Developing 以上) のオリジナル作品を制作する学習者を予測
- (モデル 2) CT スキルの合計点数が 15 点以上 (Master) のオリジナル作品を制作する学習者を予測

表 2 正例クラス、負例クラスに該当する学習者の内訳

	正例クラス	負例クラス
モデル 1	3,436	1,108
モデル 2	1,345	4,800

目的変数は、 N ($2 \leq N \leq 20$) 番目に目標習熟度に到達するオリジナル作品を制作した学習者を正例クラス、それ以外の学習者を負例クラスとする。提案するモデルでは、学習者が自身で制作したオリジナル作品が目標習熟度に到達することを予測するモデルを構築し、たとえリミックス作品で目標習熟度以上の点数を獲得していても目標習熟度に到達したと判定しない。 $N = 1$ となる正例クラスは、特微量として使用する作品数が 0 件となるため除外する。

説明変数には、過去に制作した作品における CT スキル 7 概念の 0 点から 3 点の獲得回数を用いる。獲得点数の計測は、リミックス作品の制作による学習効果が示されているため [6]、オリジナル作品の制作において獲得した点数と、リミックス作品の制作において獲得した点数は区別する。したがって、56 次元 (7 (CT スキル 7 概念) \times 4 (0 点から 3 点) \times 2 (オリジナル作品またはリミックス作品)) の説明変数を計測する。

図 6 は、構築する 2 つのモデルにおける正例クラスと負例クラスの事例を示し、3 人の学習者 (A, B, C) が 1 番目から 20 番目までに制作した作品事例から、説明変数の計測方法を述べる。

- **学習者-A:** 一度も Developing 以上のオリジナル作品を制作していないため、どちらのモデルにおいても負例クラスとなり、説明変数の計測には 1 番目から 20 番目までの全ての作品を使用する。

- **学習者-B:** 9番目の作品で Developing のオリジナル作品を制作しているため、モデル1では正例クラスとなり、学習データには1番目から8番目の作品を使用する。一方で、Master のオリジナル作品は一度も制作していないため、モデル2では負例クラスとなり、説明変数の計測には1番目から20番目までの全ての作品を使用する。
- **学習者-C:** 5番目の作品で Developing のオリジナル作品を制作しているため、モデル1では正例クラスとなり、説明変数の計測には1番目から4番目の作品を使用する。加えて、11番目の作品で Master のオリジナル作品を制作しているため、モデル2においても正例クラスとなり、説明変数の計測には1番目から10番目の作品を使用する。

予測モデルの構築には、3章で使用したデータセットを使用し、表2は、2つのモデルにおける正例クラス、負例クラスに分類された学習者の内訳を示す。

本研究では、習熟度到達予測モデルの構築にランダムフォレスト法 [13] を用いる。ランダムフォレスト法は、与えられた訓練データから複数の決定木を作成することでモデルを生成するアンサンブル学習法であり、予測結果から分類精度に強く影響した特徴量を特定できる。ランダムフォレストの実装には Python の `scikit-learn` を用いる。また、ランダムフォレストのパラメータは Breiman らが推奨する値をもとに、決定木の個数は 200、使用する特徴量の個数は説明変数 56 次元の平方根とする。また、予測モデルの検証には、層化 K-分割交差検証法を用いる。交差検証法は、データを K 個に分割し、K-1 個を学習データ、残り1個を検証データとして使用する手法である。層化 K-分割交差検証法は交差検証法の一つであり、モデル作成時に使用する学習データの割合を均等にすることで、K-分割交差検証法が抱える学習データの偏りから生じる学習不足や過学習の問題を改善する手法である。本研究では、 $K=10$ として交差検証を行う。また、予測モデルの構築時に使用する正例クラスと負例クラスに属するデータの割合は、表2に示すように偏りが存在するため、モデル構築時に各クラスの出現する割合の逆数に基づいた重みを割り当てる。予測モデルの評価指標は適合率、再現率、F 値の3つを用い、交差検証により 10 回の構築を行うことで予測モデルから得た各指標の平均値を算出する。

4.2 モデルに影響する特徴量の重要度

学習者がオリジナル作品の制作において目標習熟度に到達するまでに、過去に制作した作品の CT スキルの特徴を明らかにするために、モデルへの精度向上に寄与する各説明変数の重要度を算出する。予測モデルの評価では層化 10 分割交差検証を用いて 10 個の予測モデルを構築するため、モデルの構築毎に算出される各説明変数の重要度

表 3 2つのモデルの分類精度 (平均値)

	適合率	再現率	F 値
モデル 1	1.00	0.93	0.96
モデル 2	0.93	0.83	0.88

が異なる。本研究では、10 個のモデルにおける説明変数の重要度ランキングを明らかにするために、順位傾向のクラスタ分析手法 Scott-Knott Effect Size Difference (ESD) 検定 [14] を用いる。Scott-Knott ESD は統計的に複数のスコアをクラスターリングする手法であり、予測モデルに影響を及ぼす説明変数の特定にも使用されている。本研究では、10 個の各予測モデルから得られる説明変数の順位を分類し、いずれのモデルにも強く寄与する説明変数を特定する。Scott-Knott ESD の実装には、R パッケージの `ScottKnottESD` を用いる。

5. 実験結果

本章は、2つの習熟度到達予測モデル (モデル1: CT スキルの合計点数が 8 点以上 (Developing 以上) のオリジナル作品を制作する学習者を予測、モデル2: CT スキルの合計点数が 15 点以上 (Master) のオリジナル作品を制作する学習者を予測) の精度評価、および分類精度に寄与する説明変数を明らかにする。

5.1 習熟度到達予測モデルの分類精度

表3は、2つの予測モデルそれぞれの分類精度 (適合率、再現率、F 値) を示す。モデル1の F 値は 0.96 であり、モデル2の F 値は 0.88 であった。モデル2の分類精度はモデル1に比べて低いため、20 件の作品制作で一度も Master に到達しなかった学習者の一部は、Master に到達した学習者との間で、過去に作品制作に使用した CT スキルに違いがないことが示唆される。続いて、高い精度で分類できた2つのモデルに寄与する特徴量の重要度を分析し、さらに6章でモデルの妥当性を考察する。

5.2 特徴量の重要度

表4は、2つの予測モデルそれぞれにおいて、分類に寄与する説明変数の重要度の順位を示す。表中には、分類に強く寄与する説明変数を { 作品の種類/CT スキルの概念/点数 } のように示す。モデル1では、説明変数 { オリジナル/論理/0 点 } (オリジナル作品の制作により CT スキルの概念「論理」で 0 点を獲得した回数) が分類精度に最も寄与し、続いて { リミックス/フロー制御/2 点 }, { オリジナル/同期/0 点 } が寄与していることが分かった。また、モデル2では、{ オリジナル/論理/0 点 }, { オリジナル/同期/0 点 }, { リミックス/フロー制御/2 点 } が分類精度に寄与していることが分かった。CT スコアの概念で 1 点以上獲得している説明変数に着目すると、共通している説明

表 4 2つのモデルにおける重要度の高い説明変数の上位 10 件

グループ	モデル 1		モデル 2	
	重要度	説明変数	重要度	説明変数
1	0.10	{ オリジナル/論理/0 点 }	0.12	{ オリジナル/論理/0 点 }
2	0.07	{ リミックス/フロー制御/2 点 }	0.07	{ オリジナル/同期/0 点 }
3	0.06	{ オリジナル/同期/0 点 }	0.06	{ リミックス/フロー制御/2 点 }
4	0.06	{ リミックス/論理/0 点 }	0.06	{ オリジナル/抽象化/0 点 }
5	0.05	{ リミックス/データ表現/1 点 }	0.05	{ オリジナル/並列/0 点 }
6	0.05	{ オリジナル/データ表現/0 点 }	0.05	{ リミックス/論理/0 点 }
7	0.05	{ オリジナル/並列/0 点 }	0.05	{ リミックス/データ表現/1 点 }
8	0.05	{ オリジナル/抽象化/0 点 }	0.05	{ オリジナル/データ表現/1 点 }
9	0.04	{ リミックス/ユーザ対話性/1 点 }	0.04	{ リミックス/抽象化/1 点 }
10	0.04	{ リミックス/抽象化/1 点 }	0.03	{ オリジナル/フロー制御/1 点 }

表 5 { オリジナル/論理/0 点 } の獲得回数

	正例クラス	負例クラス
モデル 1	2 回 (± 2.0 回)	8 回 (± 8.0 回)
モデル 2	4 回 (± 3.0 回)	11 回 (± 5.5 回)

変数 { リミックス/フロー制御/2 点 }, { オリジナル/データ表現/1 点 }, { リミックス/データ表現/1 点 } がある一方, モデル 2 では { オリジナル/フロー制御/1 点 } が強く寄与していることが分かる.

6. 考察

6.1 モデルの精度に寄与する要因

本研究では, 学習者が特定の習熟度に到達するまでに制作した作品に使用する CT スキルの理解を目的に, 学習者が過去に使用した CT スキルの概念に基づき, 新たに制作する作品が特定の習熟度に到達するか否かを予測する機械学習モデルを構築した. モデルの評価実験の結果, 習熟度 Developing に関する予測 (モデル 1) の予測精度 (F 値) は 0.96, 習熟度 Master に関する予測 (モデル 2) の予測精度 (F 値) は 0.88 となり, 高い精度で分類できた. その理由は 2 つ考えられる.

- **正例クラスと負例クラスの学習者が使用した概念の違い.** 表 4 より, 2つのモデルにそれぞれ寄与する説明変数は一部順位が異なるものの, 同じ説明変数が多かった. 特に両モデルにおいて最も寄与する説明変数 { オリジナル/論理/0 点 } の獲得回数をそれぞれ比較した結果を表 5 に示す. Developing や Master に到達していない学習者は, 作品制作において概念「論理」を使用していないことが示唆される. モデル 2 において偽陽性 (負例クラスの学習者を, 誤って正例クラスの学習者と判定) に分類された学習者は 79 人存在した. これら学習者の { オリジナル/論理/0 点 } の獲得回数は, 中央値が 5 (四分位偏差が 3) と負例クラス全体の中央値よりも低いため, 20 番目までに制作する作品では Master に到達しなかったものの, 21 番目以降に制作する作品では Master に到達する可能性があることが考えられる.

- **正例クラスと負例クラスの学習に使用した作品数の違い.** 正例クラスは特定の習熟度に到達するまでの作品における CT スキル 7 概念の 0 点から 3 点の獲得回数であり, 具体的には, モデル 1 の正例クラスにおいて学習に使用した作品数の中央値は 5 件 (四分位偏差: 3.5), モデル 2 の正例クラスにおいて学習に使用した作品数は中央値 11 件 (四分位偏差: 5.5) であった. 一方, 負例クラスは特定の習熟度に到達しなかった学習者であるため分析対象とする全作品 20 件から計測している. したがって, 負例クラスの学習者が各概念の低い点数の作品を多数制作しているため, 高い精度で分類できたと考える. 特定の習熟度に到達するまでに要する時間や作品制作の経験数は, 学習者の能力や背景知識に依存する [4] ため, 今後は, 学習者が制作する作品に基づき, 特定の習熟度に到達するまでに要する作品制作数を明らかにする.

6.2 プログラミング学習時におけるモデルの活用

本研究において構築した習熟度到達予測モデルは, プログラミング学習者が有する CT スキルの把握, および Scratch において公開される作品の中から学習者の CT スキルに合わせた作品の推薦に貢献できると考える.

Scratch において, 学習者はリミックス機能などを用いて, 他の作品を参考にした作品制作を行うことも多い. 学習者がリミックス対象となる作品を選択するまでの手順は, Scratch のサービス上で自然言語による作品の検索を行い, 検索単語に紐づいた作品の抽出が挙げられる. ただし, Scratch の検索結果は, 閲覧数の多い作品やリミックス回数の多い作品など, 人気度の高い順に表示されるため, 学習者の CT スキルに合った作品が必ずしも上位に表示されるとは限らない. この課題の解決策の一つとして, Dr.Scratch の評価結果に基づく作品提示方法が考えられる. 例えば, 学習者が習熟度 Basic の作品を制作した場合には, 習熟度 Developing の作品を検索結果として提示することで, 学習者は自身の CT スキルに合った作品を見つけることが容易になる. ただし, 本研究において, 特定の

習熟度への到達可否は、学習者が過去に使用した概念の影響を受けることが明らかになったため、単に習熟度が1段階上の作品を提示するだけでは、学習者のCTスキルに合わせた作品の提示は困難であると考えられる。今後は、学習者が過去に使用したCTスキル概念に基づく、学習者のCTスキルに合わせた作品提示を検討する。

6.3 制約

本研究では、学習者が自身で制作するオリジナル作品が目標習熟度に到達することを予測するモデルの構築を行ったが、オリジナル作品の中には、公開されている他の作品を模倣して制作した作品が存在することも考えられる。また、Scratchのサービス上に共有されている作品しか調査できていないため、サービス上に非公開の作品を制作していることも考えられる。本研究で対象とする学習者が制作した作品の中にも、これらの事例が含まれる場合、予測モデルの精度低下に繋がることが考えられる。

7. おわりに

本研究では、まず事前分析として、学習者が制作した作品のCTスキルの合計点数を調査した結果、学習者は突如点数の高い作品を制作することはなく、類似のCTスキルを使用した作品を連続で制作していることが示唆された。その後、学習者が制作する作品を3つの習熟度(Basic, Developing, Master)に分類し、作品制作に使用するCTスキル概念を分析した。具体的には、学習者が過去に使用した概念に基づき、新たに制作する作品がDeveloping以上、またはMasterの習熟度に到達するか否かを予測する2つの習熟度到達予測モデルを構築した。その結果、Developing以上に関する予測の精度(F値)は0.96、習熟度Masterに関する予測の精度(F値)は0.88と高い精度で分類することができた。分類精度が高くなった要因として、目標習熟度に到達する学習者と到達しない学習者の間では使用する概念(「論理」など)や、モデルの学習に使用した作品数に違いがあることが考えられる。特に、モデルの学習に使用した作品数の違いについては、目標習熟度に到達するまでに必要とする作品制作回数は学習者によって異なることが、本研究により明らかとなった。今後は、学習者各自が特定の習熟度に到達するまでに必要とする作品制作回数を明らかにし、学習者のCTスキル習熟速度も考慮した習熟度到達予測モデルを提案する。

参考文献

[1] Wing, J. M.: Computational thinking and thinking about computing, *Journal of Philosophical Transactions of the Royal Society*, Vol. 366, pp. 3717–3725 (2018).
[2] Moreno-León, J., Robles, G. and Román-González, M.: Dr. Scratch: Automatic analysis of scratch projects to

assess and foster computational thinking, *RED. Revista de Educación a Distancia*, No. 46, pp. 1–23 (2015).
[3] Moreno-León, J. and Robles, G.: Analyze your Scratch projects with Dr. Scratch and assess your computational thinking skills, *Scratch conference*, pp. 12–15 (2015).
[4] Yang, S., Domeniconi, C., Revelle, M., Sweeney, M., Gelman, B. U., Beckley, C. and Johri, A.: Uncovering Trajectories of Informal Learning in Large Online Communities of Creators, *In Proceedings of the 2nd Conference on Learning @ Scale (L@S'15)*, pp. 131–140 (2015).
[5] Weintrop, D. and Wilensky, U.: Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms, *Transactions on Computing Education*, Vol. 18, No. 1, pp. 1–25 (2017).
[6] Dasgupta, S., Hale, W., Monroy-Hernández, A. and Hill, B. M.: Remixing As a Pathway to Computational Thinking, *Proceedings of the 19th Conference on Computer-Supported Cooperative Work & Social Computing (CSCW'16)*, pp. 1438–1449 (2016).
[7] 杉浦学, 松澤芳昭, 岡田健, 大岩元: アルゴリズム構築能力育成の導入教育: 実作業による概念理解に基づくアルゴリズム構築体験とその効果, *情報処理学会論文誌*, Vol. 49, No. 10, pp. 3409–3427 (2008).
[8] Aivaloglou, E., Hermans, F., Moreno-León, J. and Robles, G.: A Dataset of Scratch Programs: Scraped, Shaped and Scored, *Proceedings of the 14th International Conference on Mining Software Repositories (MSR'17)*, pp. 511–514 (2017).
[9] Robles, G., Moreno-León, J., Aivaloglou, E. and Hermans, F.: Software clones in scratch projects: on the presence of copy-and-paste in computational thinking learning, *Proceedings of the 11th International Workshop on Software Clones (IWSC'17)*, pp. 1–7 (2017).
[10] Troiano, G., Snodgrass, S., Argimak, E., Robles, G., Smith, G., Cassidy, M., Tucker-Raymond, E., Puttick, G. and Hartevelde, C.: Is My Game OK Dr. Scratch?: Exploring Programming and Computational Thinking Development via Metrics in Student-Designed Serious Games for STEM, *Proceedings of the 18th International Conference on Interaction Design and Children (IDC'19)*, pp. 208–219 (2019).
[11] Troiano, G., Chen, Q., Vargas-Alba, Á., Robles, G., Smith, G., Cassidy, M., Tucker-Raymond, E., Puttick, G. and Hartevelde, C.: Exploring How Game Genre in Student-Designed Games Influences Computational Thinking Development, *Proceedings of the Conference on Human Factors in Computing Systems (CHI'20)*, pp. 1–17 (2020).
[12] Aivaloglou, E. and Hermans, F.: How Kids Code and How We Know: An Exploratory Study on the Scratch Repository, *Proceedings of the Conference on International Computing Education Research (ICER'16)*, pp. 53–61 (2016).
[13] Breiman, L.: Random forests, *Machine learning*, Vol. 45, No. 1, pp. 5–32 (2001).
[14] Tantithamthavorn, C., McIntosh, S., Hassan, A. E. and Matsumoto, K.: The Impact of Automated Parameter Optimization on Defect Prediction Models, *IEEE Transactions on Software Engineering*, Vol. 45, No. 7, pp. 683–711 (2019).