

OpenLA

ラーニングアナリティクス推進のための
オープンソースライブラリ

島田敬士

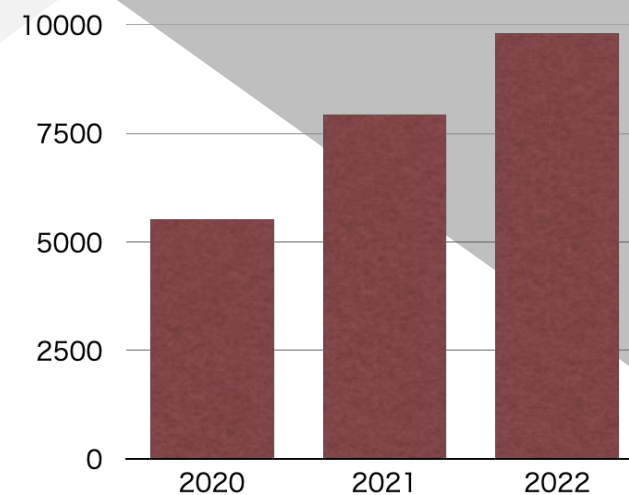
九州大学大学院システム情報科学研究院



KYUSHU UNIVERSITY

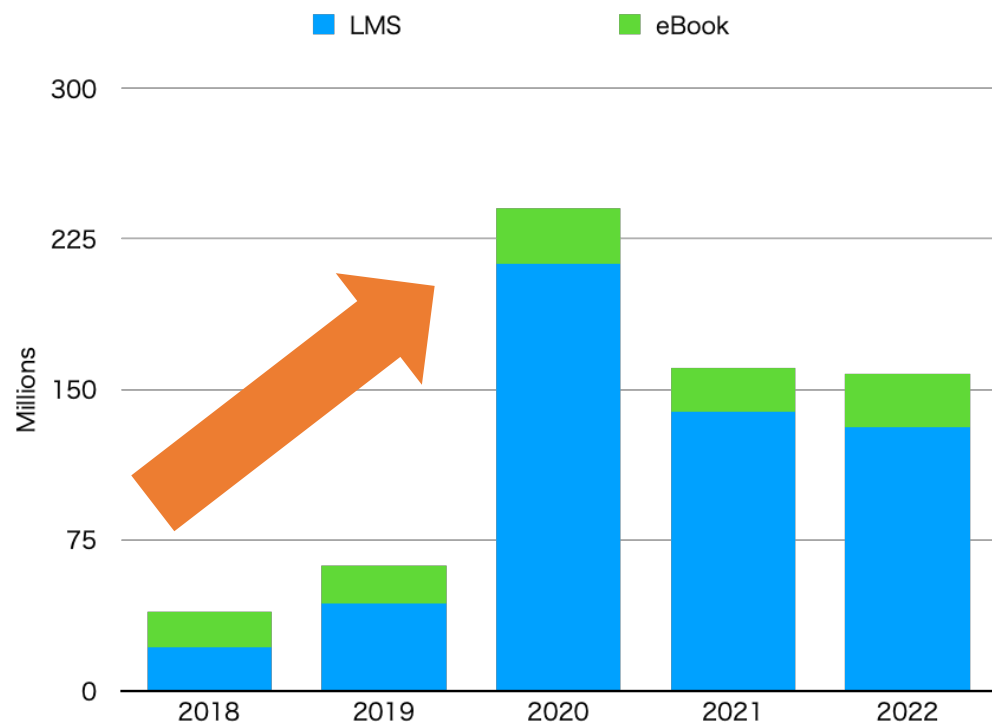
Released in June 2020

downloads 27k

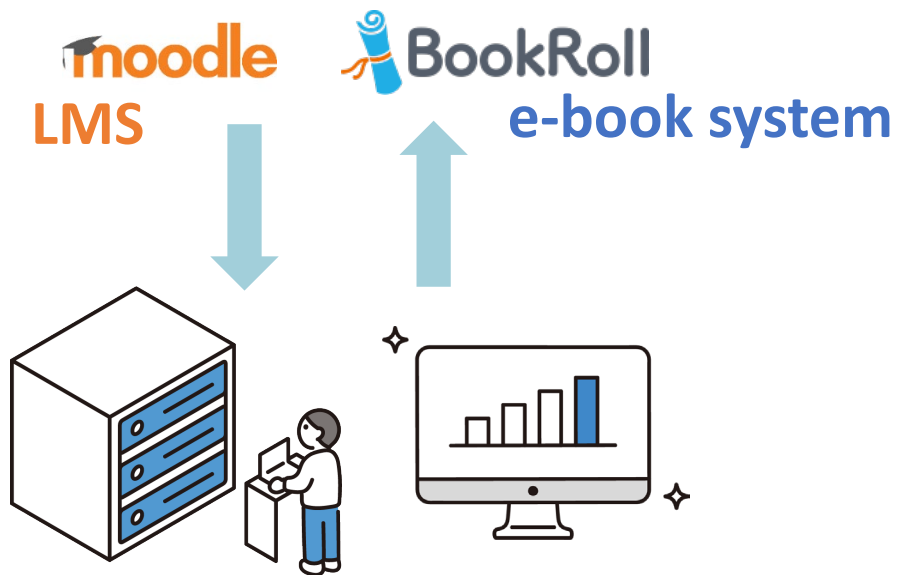
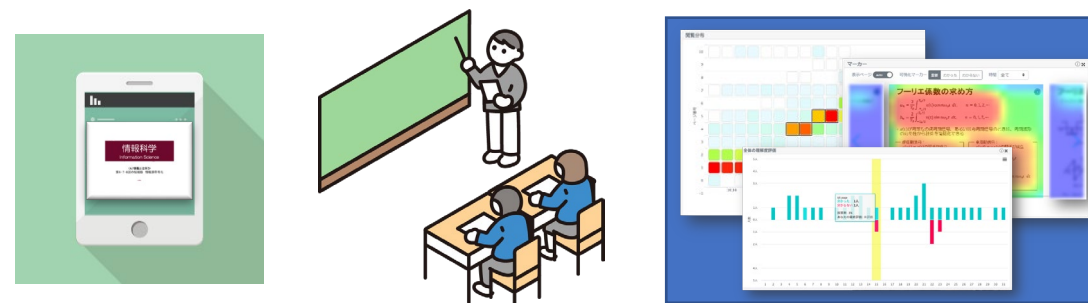


社会背景

コロナ禍以降、ICT利用教育が浸透
教育データの蓄積が容易



データ駆動型教育への期待



そもそも、教育データはどのように記録されている？

e-book system

User id	Contents id	Operation name	Page	Event time
A	X	OPEN	1	2020/4/10 14:30:31
B	X	OPEN	1	2020/4/10 14:31:40
A	X	NEXT	1	2020/4/10 14:31:50
⋮	⋮	⋮	⋮	⋮

「いつ」、「どこで」、「誰が」、「何を」した？

LMS

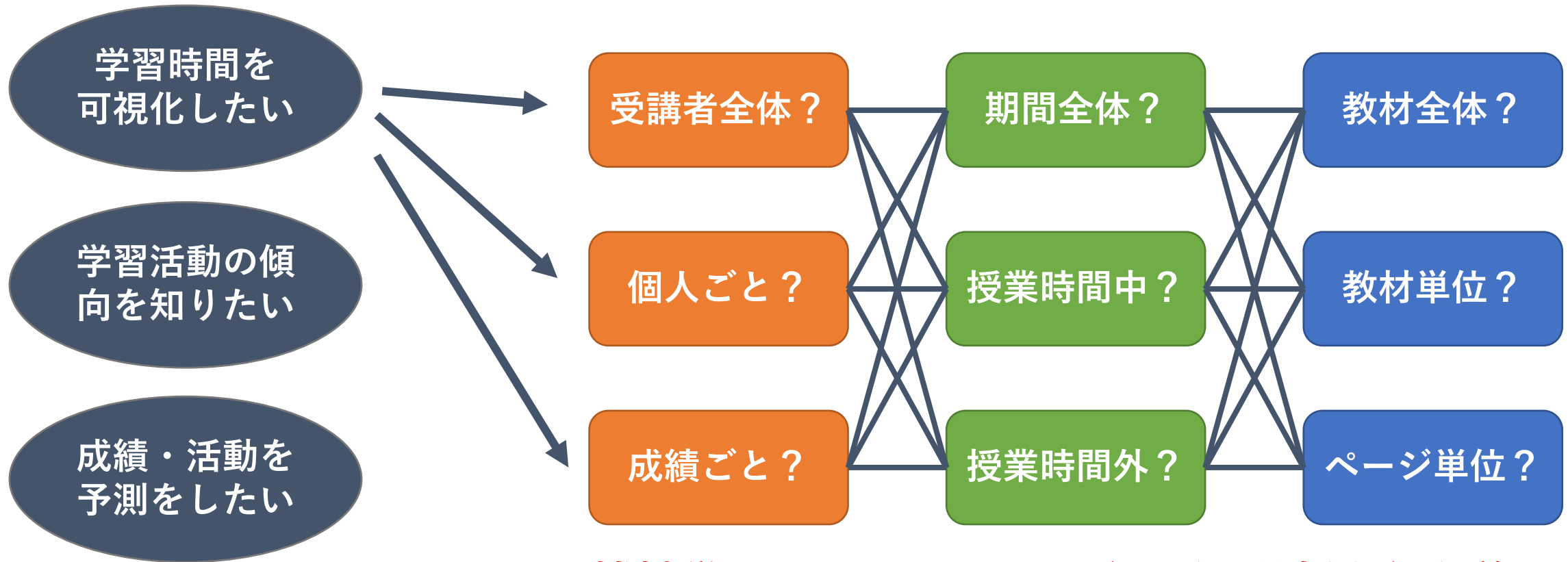
講義日，講義時間帯：datetime 型

小テストのスコア，最終成績：int/float 型

教育データ分析の目的・方法は？

目的は様々・・・

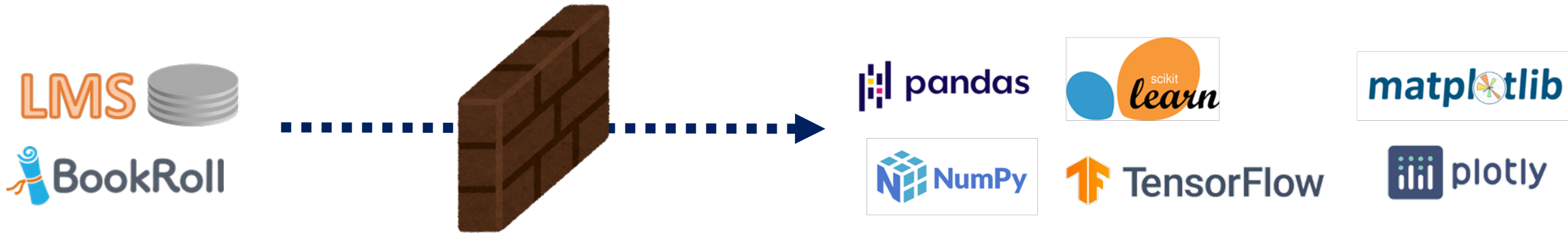
分析対象者，期間，教材等も様々・・・



機械学習やクラスタリングなどの分析を行う前の
データ抽出，整形等の事前処理がそこそこ面倒

教育データ活用に向けた課題

- 事前処理が意外に大変
- これまでは、研究者、技術者が個別に事前処理部のコードを開発
- 事前処理は後続処理に比べると共通部分が多い → 効率性を上げられないか？

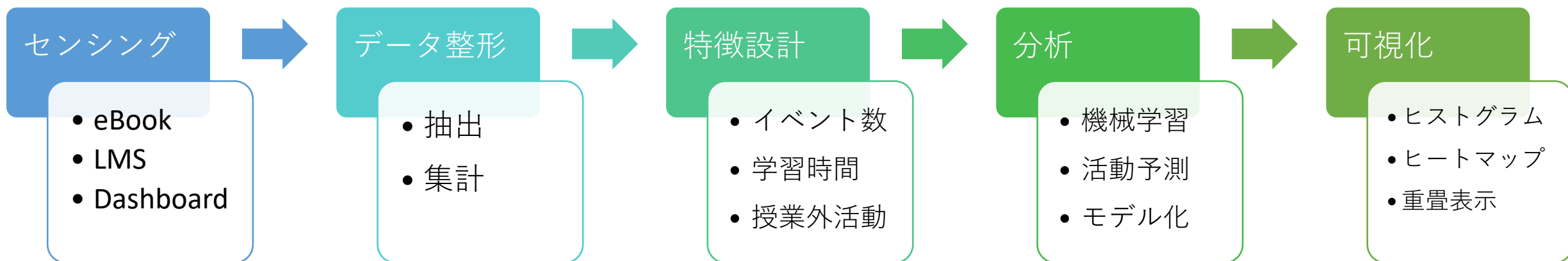


- データの読み出し処理
- 分析対象のデータの抽出処理
- 集計や変換処理
- 結果の可視化处理

OpenLAの役割

データソースと特徴計算，機械学習等のライブラリ間のミッシングリンクを補完
プログラミング効率を向上

ライブラリ，チュートリアル資料が充実
様々な実装事例とともに公開



BookRoll

LMS



OpenLA

pandas

NumPy

scikit-learn

TensorFlow

matplotlib

plotly

OpenLAのモジュール群

データセット

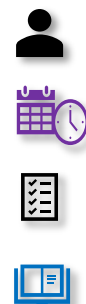


- EventStream
- LectureMaterial, LectureTime
- QuizScore (GradePoint)

Course Information Module

コースに関する情報を取得

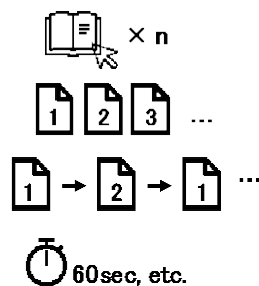
- 登録ユーザ
- 講義時間
- 最終成績
- イベントログ



Data Conversion Module

イベントログを集計・変換

- 各操作の回数を集計
- ページごとに集計
- ページ遷移ごとに集計
- 特定の時間幅ごとに集計



Data Extraction Module

ログから必要な情報を抽出

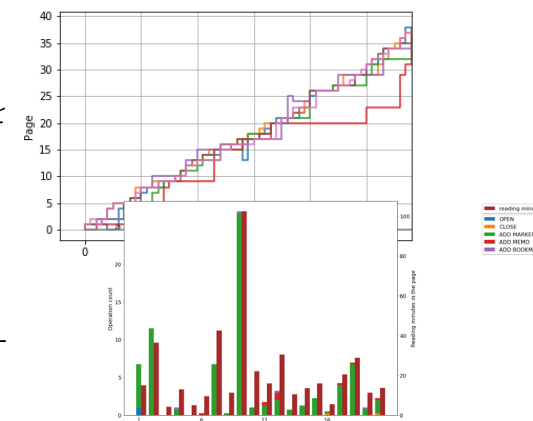
- 成績の高い学生など特定のユーザのログ
- 分析したいタイミングのログ...



Data Visualization Module

データの可視化

- 学生がどのページを読んでいたか (1分ごとに集計)
- 授業資料の各ページにおける閲覧時間やマーカーの個数...



Data Conversion Moduleの利用例①

> 操作回数の集計

User id	Contents id	Operation name	Page	Event time
A	X	OPEN	1	2020/4/10 14:30:31
B	X	OPEN	1	2020/4/10 14:31:40
A	X	NEXT	1	2020/4/10 14:31:50
⋮	⋮	⋮	⋮	⋮

convert_into_operation_count関数

各学生の各コンテンツに対する操作数

User id	Contents id	OPEN	CLOSE	NEXT	PREV	MARKER	MEMO	...
A	X	2	2	64	16	4	32	...
B	X	4	4	128	64	32	8	...
C	X	1	1	32	8	0	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
A	Y	1	1	32	8	8	2	...
B	Y	2	2	64	32	16	4	...
C	Y	1	1	16s	4	2	2	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Data Conversion Moduleの利用例②

> ページごとの 学習活動の集計

User id	Contents id	Operation name	Page	Event time
A	X	OPEN	1	2020/4/10 14:30:31
B	X	OPEN	1	2020/4/10 14:31:40
A	X	NEXT	1	2020/4/10 14:31:50
⋮	⋮	⋮	⋮	⋮

convert_into_page_wise関数

各ページの閲覧時間と操作数

User id	Contents id	Page	Reading seconds	MARKER	MEMO	⋮
A	X	1	64	0	0	⋮
A	X	2	128	1	0	⋮
A	X	3	1024	5	2	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮
B	X	1	81	0	0	⋮
B	X	2	729	4	1	⋮
B	X	3	243	0	1	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Data Conversion Moduleの利用例③

> ページ遷移ごとの 学習活動の集計

User id	Contents id	Operation name	Page	Event time
A	X	OPEN	1	2020/4/10 14:30:31
B	X	OPEN	1	2020/4/10 14:31:40
A	X	NEXT	1	2020/4/10 14:31:50
⋮	⋮	⋮	⋮	⋮

convert_into_page_transition関数

各ページの閲覧時間と操作数
(ページ遷移ごと)

User id	Contents id	Page	Reading seconds	MARKER	MEMO	⋮
A	X	1	210	3	2	⋮
A	X	2	63	1	0	⋮
A	X	1	42	1	1	⋮
A	X	3	490	0	2	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Data Conversion Moduleの利用例④

> 単位時間ごとの 学習活動の集計

User id	Contents id	Operation name	Page	Event time
A	X	OPEN	1	2020/4/10 14:30:31
B	X	OPEN	1	2020/4/10 14:31:40
A	X	NEXT	1	2020/4/10 14:31:50
⋮	⋮	⋮	⋮	⋮

convert_into_time_range関数

各区間で最も長く閲覧されたページと
その区間の操作

User id	Contents id	Elapsed seconds	Page	MARKER	MEMO	⋮
A	X	60	1	1	0	⋮
A	X	120	2	2	1	⋮
A	X	180	2	2	0	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮
B	X	60	1	0	0	⋮
B	X	120	4	0	0	⋮
B	X	180	5	1	0	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮

コード量削減例

without OpenLA

```
import datetime
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from matplotlib import ticker as tick
lecture_week = 1
event_stream = pd.read_csv("dataset_sample/Course_A_EventStream.csv")
lecture_schedule = pd.read_csv("dataset_sample/Course_A_LectureTime.csv")
contents_information = pd.read_csv("dataset_sample/Course_A_LectureMaterial.csv")
contents_id = contents_information[contents_information["lecture"] == lecture_week]
lecture_schedule["starttime"] = pd.to_datetime(lecture_schedule["starttime"])
lecture_schedule["endtime"] = pd.to_datetime(lecture_schedule["endtime"])
lecture_start_time = lecture_schedule[lecture_schedule["lecture"] == lecture_week]
lecture_end_time = lecture_schedule[lecture_schedule["lecture"] == lecture_week]
event_stream["eventtime"] = pd.to_datetime(event_stream["eventtime"])
event_stream = event_stream[event_stream["contentsid"] == contents_id]
event_stream = event_stream[lecture_start_time < event_stream["eventtime"]]
event_stream = event_stream[event_stream["eventtime"] < lecture_end_time - datetime.timedelta(minutes=15)]
prev_page_dict = dict(zip(event_stream["userid"], np.zeros(len(event_stream))))

seconds = [sec for sec in range(0, (lecture_end_time - lecture_start_time - datetime.timedelta(minutes=15)).seconds, 60)]
plt.figure()
max_time_in_figure = 0
max_page_in_figure = 0
for user_id, user_stream in event_stream.groupby('userid'):
    prev_page = 0
    elapsed_minutes_list = []
    pages_list = []
    user_stream.reset_index()
    for sec in seconds:
        start_of_range = lecture_start_time + datetime.timedelta(seconds=sec)
        end_of_range = start_of_range + datetime.timedelta(seconds=60)
        stream_in_range = user_stream[(user_stream['eventtime'] < end_of_range) &
                                     (start_of_range < user_stream['eventtime'])]
    if stream_in_range.empty:
        longest_staying_page = prev_page
    else:
        pages = stream_in_range['pageno']
        event_time = stream_in_range['eventtime']
        staying_time_dict = (dict(zip(pages.unique(), np.zeros(len(pages.unique()))))
                             + dict(zip(event_time.unique(), np.zeros(len(event_time.unique())))))
        staying_time_dict[prev_page] = (event_time.iat[0] - start_of_range).seconds
        longest_staying_time = (event_time.iat[0] - start_of_range)
        longest_staying_page = prev_page
        for idx, page in enumerate(pages[:-1]):
            if idx + 1 < len(stream_in_range):
                staying_time = event_time.iat[idx + 1] - event_time.iat[idx]
            else:
                staying_time = end_of_range - event_time.iat[idx]
            if page == pages.iat[idx + 1]:
                staying_time_dict[page] += staying_time.seconds
            else:
                staying_time_dict[pages.iat[idx + 1]] += staying_time.seconds
            if longest_staying_time < staying_time:
                longest_staying_page = page
        prev_page = longest_staying_page
        elapsed_minutes_list.append(sec/60.0)
        pages_list.append(longest_staying_page)
    plt.step(elapsed_minutes_list, pages_list)
    max_time_in_figure = max(max_time_in_figure, max(elapsed_minutes_list))
    max_page_in_figure = max(max_page_in_figure, max(pages_list))

plt.xlabel = "minutes"
plt.ylabel = "page"
plt.grid(axis='both', which='both')
plt.show()
```

14 lines

34 lines

10 lines

データ読み出し
分析対象データを抽出

4 lines

8 lines

1分ごとに学習活動ログを集計

5 lines

学生の学習活動を可視化

with OpenLA

```
import datetime
import OpenLA as la
from matplotlib import pyplot as plt

course_info, event_stream = la.start_analysis(files_dir="dataset_sample", course_id="A")

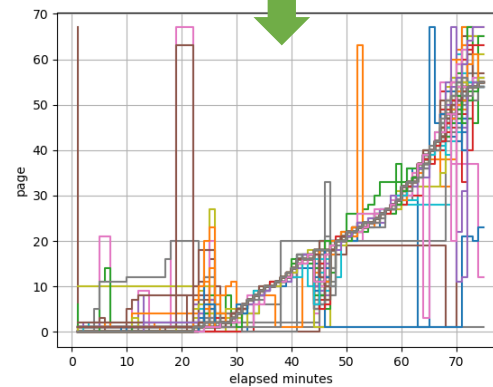
lecture_week = 1
lecture_end_time = course_info.lecture_end_time(lecture_week)
content = course_info.lecture_week_to_contents_id(lecture_week)

behavior = la.convert_into_time_range(course_info, event_stream,
                                     contents_id=content,
                                     lecture_week=lecture_week,
                                     interval_seconds=60,
                                     start_time='start_of_lecture',
                                     end_time=lecture_end_time-datetime.timedelta(minutes=15),
                                     time_range_basis='minutes',
                                     count_operation=False)

ax = la.visualize_pages_in_time_range(behavior,
                                     contents_id=content,
                                     user_id=behavior.user_id())

plt.grid(axis='both', which='both')
plt.show()
```

約 6割
削減



第7回 IMS Japan賞 優秀賞

『OpenLA：教育データ分析のためのオープンソースライブラリ』

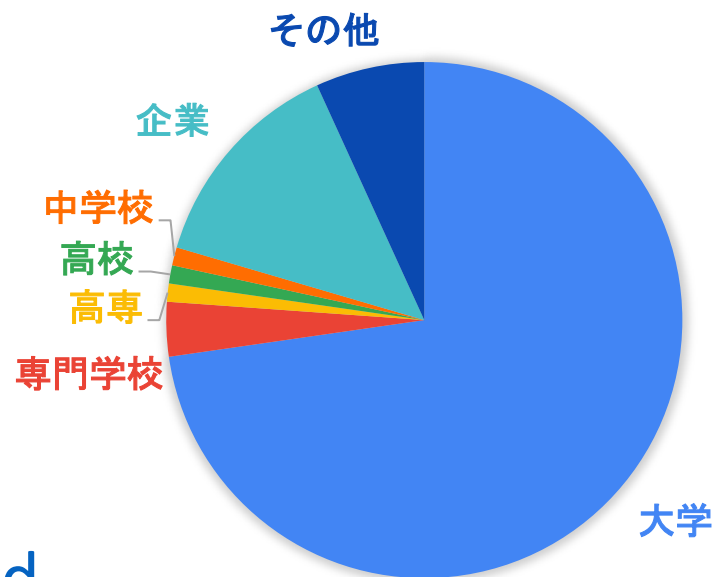


教育データ分析コンテスト（EDE主催，IPJSJ CLE研究会協賛）



イベントを通して教育データの分析技術を社会全体で向上させていくことを目的

例年，約100チームが参加



概要

教育の情報化が加速し、学習管理システム（LMS）やデジタル学習教材を利用する機会が急激に増えてきています。そのようなシステムに蓄積される教育データを活用したデータ駆動型教育の実現にも近年期待が高まっております。一方で、教育データの解析手法や活用法についてはまだまだ研究段階であり、実際にどのようなデータが学習者から収集され、どのような解析ができるのかなどのノウハウについては、あまり情報が共有されていないのが現状です。

そこで本コンテストでは、実際の教育現場で収集されたログデータを参加者に提供し、実際の教育データを分析していただき、その結果の精度や分析方法あるいは分析の着眼点の斬新さについて参加者間で競争、共有をしていただくコンテストを開催することにいたしました。特に本コンテストでは、2020年のCOVID-19の流行に伴うオンライン授業期間中の教育データと、それ以前の対面授業期間中の教育データの両方を提供することで、授業時の学習活動時の分析だけではなく、対面授業とオンライン授業の比較分析も行うことができます。また、提供する教育データはCSVフォーマットで記述されており、そのデータの読み込みや、抽出、集計、可視化などを行う基本処理関数群OpenLAも提供しております。OpenLAはPythonで記述されたライブラリですので、他のPython系のライブラリとも親和性が高く、例えばscikit-learnによる機械学習やPlotlyによる高度な描画などを行うライブラリにも簡単に処理結果を接続することが可能です。ぜひ、この機会に教育データ分析コンテストを通して教育データに触れていただき、より多くの方々と教育データ活用の可能性について共有いただければ幸いです。

参加者（個人またはチーム）は、教育データ分析の着眼点、分析手法、得られた結果を、主催者側が後日指定する形式で投稿していただけます。投稿いただいた内容を評価員により審査を行い、上位者を2022年3月に開催予定のシンポジウムで発表・表彰いたします。入賞者には賞状と副賞が贈呈される予定ですので、奮ってご参加ください。

今年度は第3回コンテストが開催中

<https://sites.google.com/view/ede-datachallenge-3rd>

昨年度のコンテストの結果：成績予測部門

目的

デジタル教材の閲覧ログと最終テストの成績データから、最終成績の予測を行うモデルを開発

	投稿者	予測誤差	利用技術
 1位	ham***	16.05	LightGBM系
 2位	leel***	18.40	Deep Learning系
3位	fa2***	19.28	LightGBM系
4位	ior***	20.32	Neural Network系
5位	fa2***	21.82	LightGBM系

ページ滞在時間やアクセス頻度、教材操作イベント集計などの特徴を各参加者が工夫しながら利用

昨年度のコンテストの結果：エビデンス発見部門

目的

2019年度（コロナ前），2020年度（コロナ禍）の授業時のデジタル教材の閲覧ログと成績データからコロナ前後の学習活動の違いを分析



復習のタイミングや量と成績の関連性についての分析



予習・復習時間に関する諸検討-実態把握から成績不振学生の予測まで-




勉強時間の定義や、資料独立タイプ、既学習タイプなどの分類

各受賞者の分析結果をコンテストWebサイトで公開中

<https://sites.google.com/view/ede-datachallenge-23>

LLM×OpenLAによる教育データ分析

 **OpenAI**
GPT-3.5

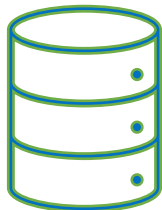
今回はGPT-3.5
を利用


LangChain

LLMの機能拡張
ライブラリ群

OpenLA

教育データの
前処理に利用



eBookの閲覧データ
教材基礎データ
成績データ

ある講義1年（1期）分のデータ

LMS


OpenLA

教育データと各種データ分析
／機械学習ライブラリ間の
ミッシングリンクを解消

 pandas



 NumPy

 TensorFlow

4 lines

データ読込
データ抽出

8 lines

データ統合

5 lines

可視化

```
import datetime
import OpenLA as la
from matplotlib import pyplot as plt

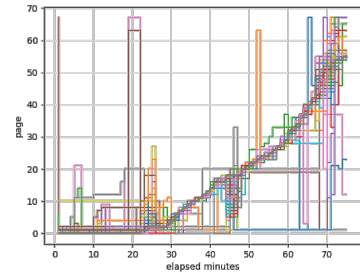
course_info, event_stream = la.start_analysis(files_dir="dataset_sample", course_id="A")

lecture_week = 1
lecture_end_time = course_info.lecture_end_time(lecture_week)
content = course_info.lecture_week_to_contents_id(lecture_week)

behavior = la.convert_into_time_range(course_info, event_stream,
                                     contents_id=content,
                                     lecture_week=lecture_week,
                                     interval_seconds=60,
                                     start_time='start_of_lecture',
                                     end_time=lecture_end_time-datetime.timedelta(minutes=15),
                                     time_range_basis='minutes',
                                     count_operation=False)

ax = la.visualize_pages_in_time_range(behavior,
                                     contents_id=content,
                                     user_id=behavior.user_id())

plt.grid(axis='both', which='both')
plt.show()
```



従来比

60%
削減

教育データ分析プログラムの開発
LAシステム開発を加速化

元々は人のコーディングプロセスを助長
するために開発されたライブラリ

例

Prompt

学生の成績を向上させる方法を見つけるために提供されたデータを詳細に分析せよ。

Response

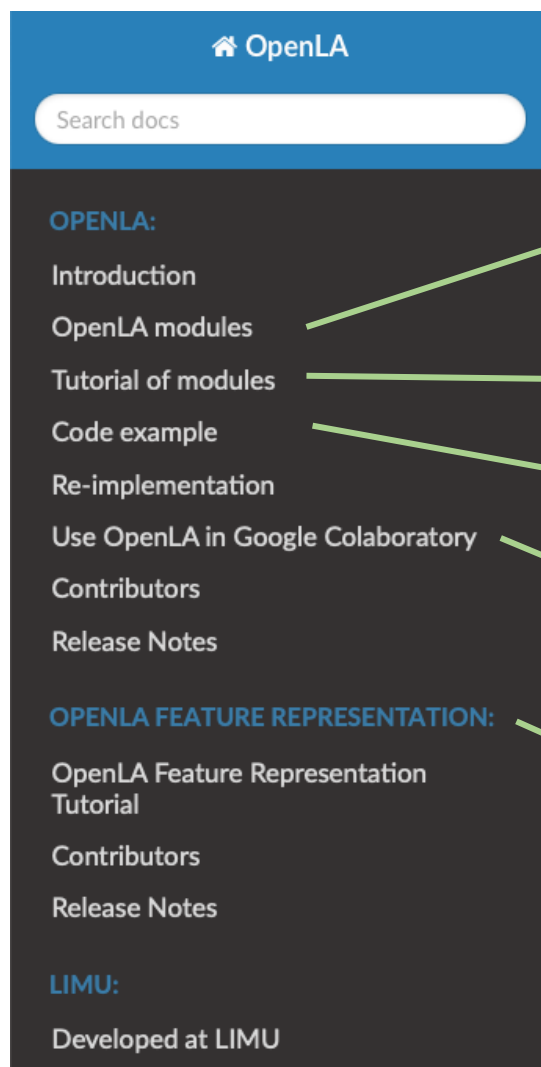
成績の良い学生ほど、マーカー、メモ、ブックマークが多く、また閲覧ページ数も多い傾向がある。このことから、マーカー、メモ、ブックマークを増やしたり、学生が教材に触れる機会を増やすことが、成績向上につながる可能性がある。

GPT-4でさらに精度が向上することも確認

OpenLAを利用することでデータの解釈性が向上？

分析成功率
OpenLAあり OpenLAなし

			分析成功率 OpenLAあり	OpenLAなし
分析レベル1 基本処理	データ数集計 データ解釈 教材ページ数 各イベント数	閲覧時間分析 各学生の活動 全ページ読破 学生数, など	9/10	4/10
分析レベル2 条件付処理	ある教材の最初の10ページの平均閲覧時間は？ 全教材で2分以上閲覧されたページは何ページ？, など		7/10	2/10
分析レベル3 パターン特定	教材の閲覧時間と学生の成績には、相関はある？ 教材の閲覧時間が平均閲覧時間を20%以上、上回った学生の人数は？		5/10	2/10
分析レベル4 モデリング	成績を予測するモデルを作成することは可能か？ 閲覧パターンや成績に基づいて教材推薦するモデルを作成可能か？		3/10	0/10
分析レベル5 高度なRQ	効果の高い教材閲覧方略や介入方法を提案できるか？ 学習効率を最大化するページの閲覧順序や学習方法を特定できるか？		7/10	4/10



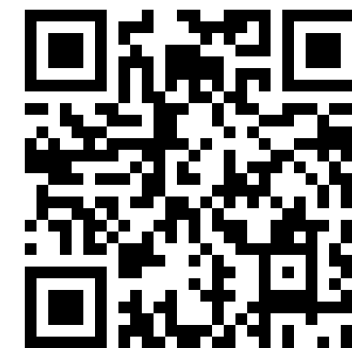
モジュール・クラス・関数の詳細

モジュールの概要

クラス・関数の具体的な利用例

Google ColabでのOpenLA利用

特徴表現作成機能



特徴表現の獲得関数群をNew Release！！

> 特徴表現作成機能

```
pip install openla-feature-representation
```

- 学習者ごとの特徴ベクトルをイベントログから作成する拡張機能
- ドキュメントは今後改訂予定

> ALP

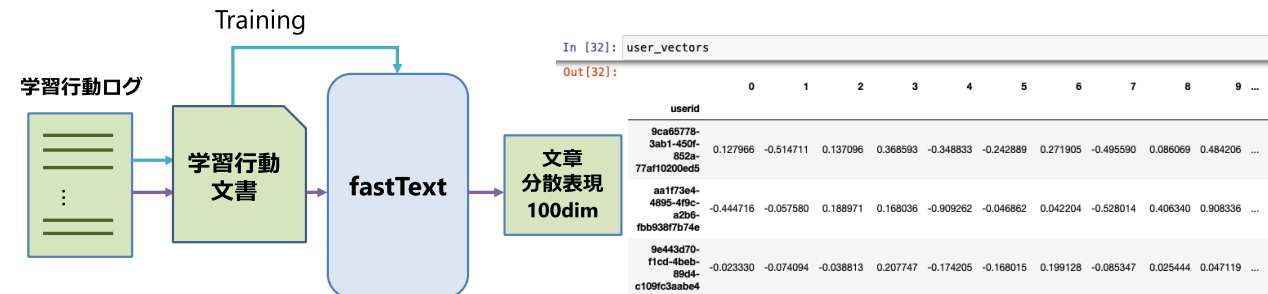
- 各週の各活動を0～5点にポイント化

Activities	5	4	3	2	1	0
Slide Views	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	o.w.
Markers	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	o.w.
Memos	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	o.w.
Actions	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	o.w.

- Fumiya Okubo, Takayoshi Yamashita, Atsushi Shimada, Hiroaki Ogata
A Neural Network Approach for Students' Performance Prediction
The 7th International Conference on Learning Analytics & Knowledge Understanding, 2017.03

> E2Vec

- e-book閲覧系列の分散表現



- 宮崎 佑馬, 峰松 翼, 谷口 雄太, 大久保 文哉, 島田 敬士
教育データの分散表現生成手法の提案とAt-risk学生検知への応用
第40回教育学習支援情報システム研究発表会 (CLE40), 2023.06

ご清聴ありがとうございました

E-mail: atsushi@ait.kyushu-u.ac.jp

URL: <http://limu.ait.kyushu-u.ac.jp/>



KYUSHU UNIVERSITY