



ラーニングアナリティクス基盤 ユーザマニュアル

国立情報学研究所

ラーニングアナリティクス基盤 ユーザマニュアル

目 次

1. 基本操作

1.1	JupyterHubの操作	3p
1.2	Supersetの操作	10p

2. 実践操作

2.1	動画の視聴回数を調べる	17p
2.2	小テストの結果から成績優秀者を特定する	21p
2.3	小テストの結果と動画の視聴履歴から退学者を予測する	27p

1. 基本操作

1.1 JupyterHubの操作

1.1.1	JupyterHubへログインする	3p
1.1.2	JupyterHubからログアウトする	6p
1.1.3	notebookを利用する	7p

1.2 Supersetの操作

1.2.1	Supersetへログインする	10p
1.2.2	Supersetからログアウトする	12p
1.2.3	Supersetを利用する	13p

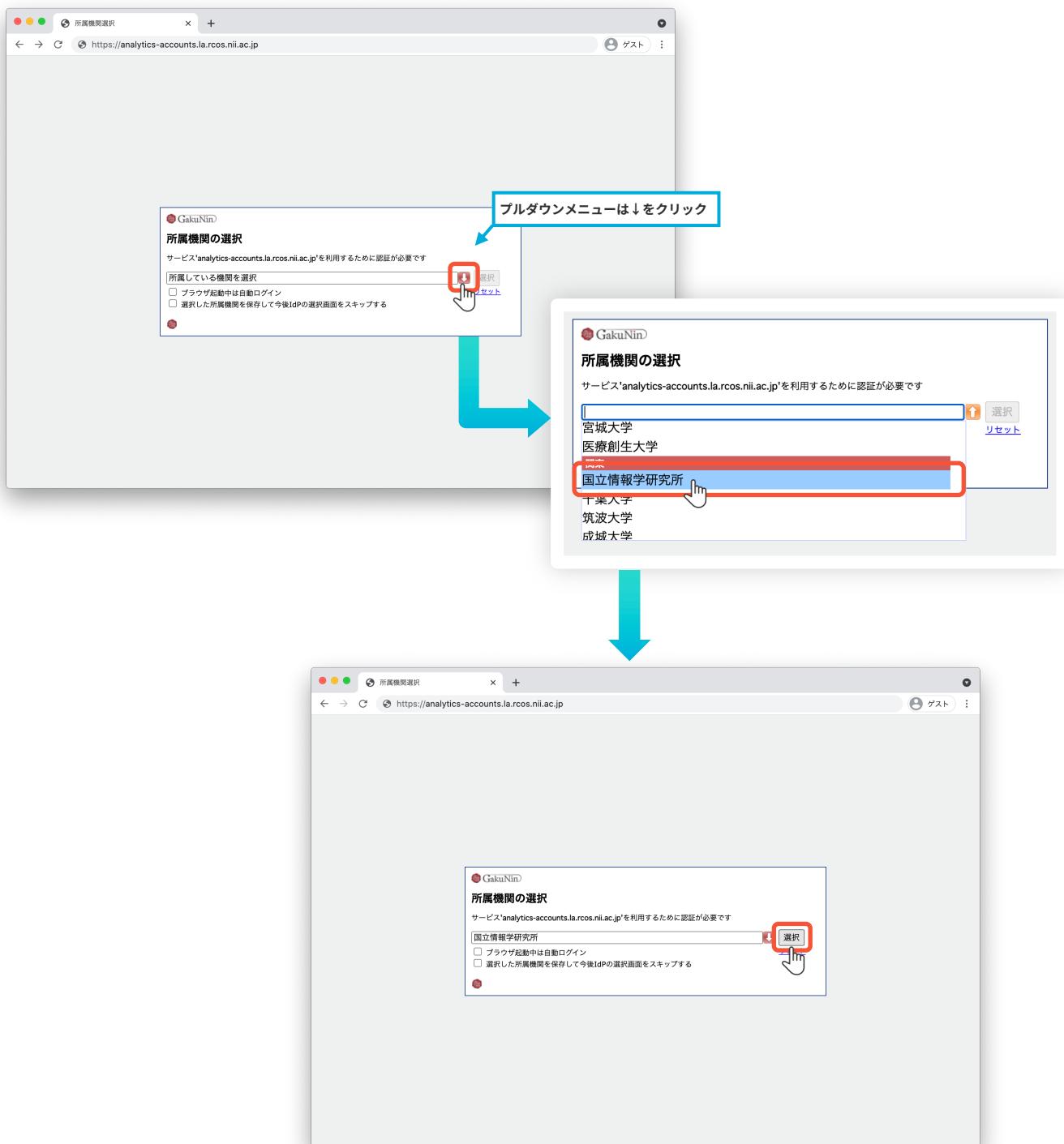
1. 基本操作 > 1.1 JupyterHubの操作 > 1.1.1 JupyterHubへログインする

1.1.1 | JupyterHubへログインする

JupyterHubとは

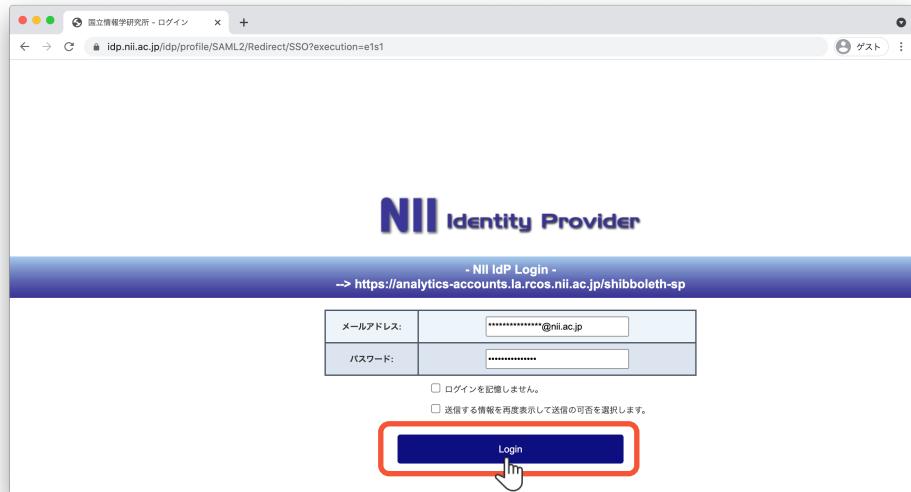
管理者が用意したPythonチュートリアル用 Jupyter notebookを、ユーザーがブラウザから実行できるようにするもの。

- ▶ <https://analytics-accounts.la.rcos.nii.ac.jp>へアクセスし、自分が所属する機関名を選択する。

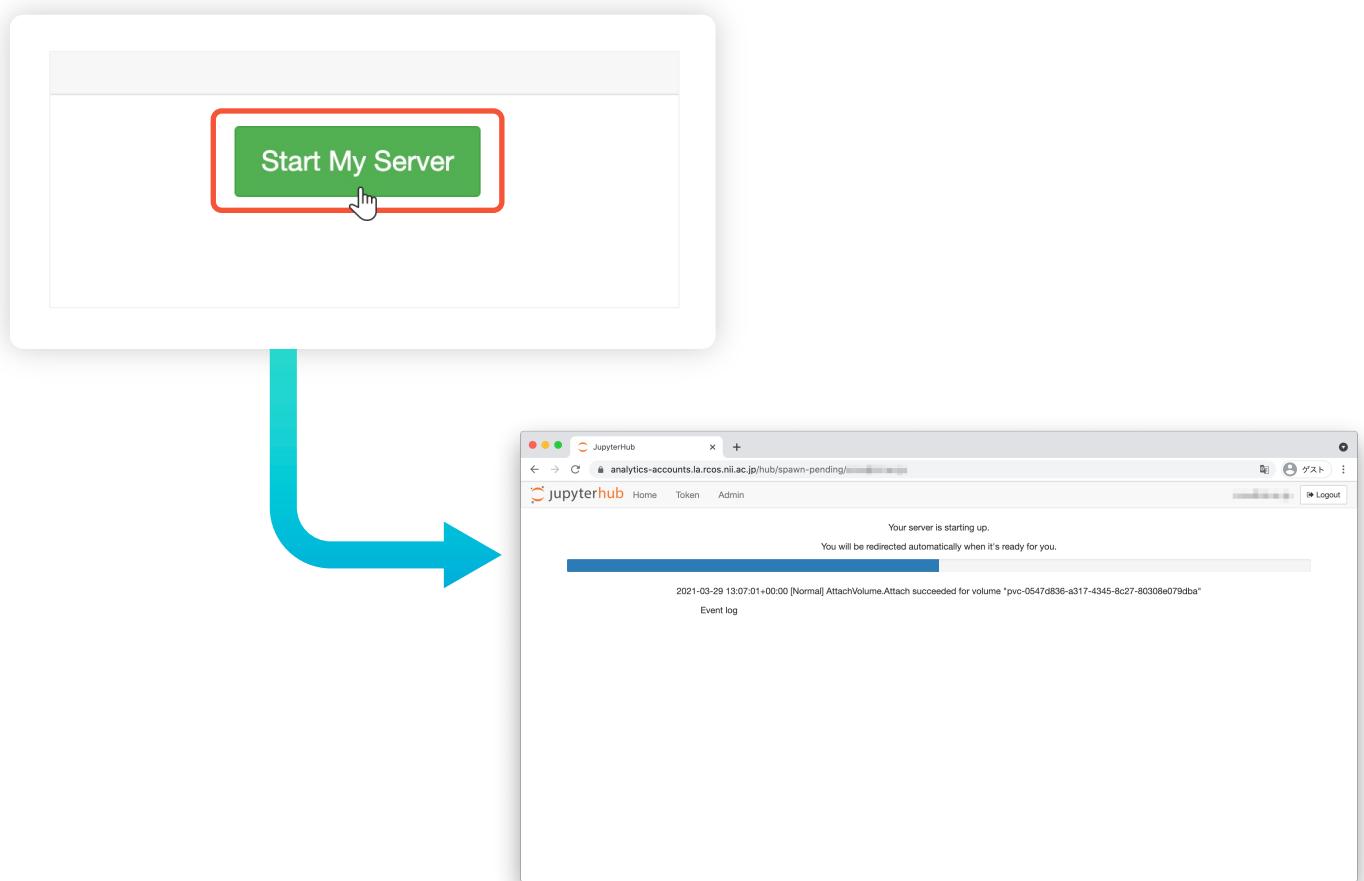


1. 基本操作 > 1.1 JupyterHubの操作 > 1.1.1 JupyterHubへログインする

- ▶ 選択した所属機関のログインページに遷移するので、付与されているメールアドレスとパスワードを入力し[Login]ボタンをクリックする。



初回アクセス時など、Jupyterノートブックサーバが起動していない場合には[Start My Server]ボタンが表示される。この場合には、ボタンをクリックしサーバを起動する。



1. 基本操作 > 1.1 JupyterHubの操作 > 1.1.1 JupyterHubへログインする

- ▶ サーバ起動後に、ノートブックの一覧画面が表示される。

The screenshot shows the JupyterHub interface with the title 'jupyterhub'. At the top right are buttons for 'Quit', 'Logout', and 'Control Panel'. Below the title are tabs for 'Files' (selected), 'Running', and 'Clusters'. A message 'Select items to perform actions on them.' is displayed above a list of files. The list includes three notebooks: 'get_caliper_statements.ipynb', 'get_xapi_statements.ipynb', and 'who_achieved_excellent_results.ipynb'. Each entry has a checkbox, a preview icon, the file name, a timestamp, and a file size. There are also buttons for 'Upload', 'New', and a refresh icon.

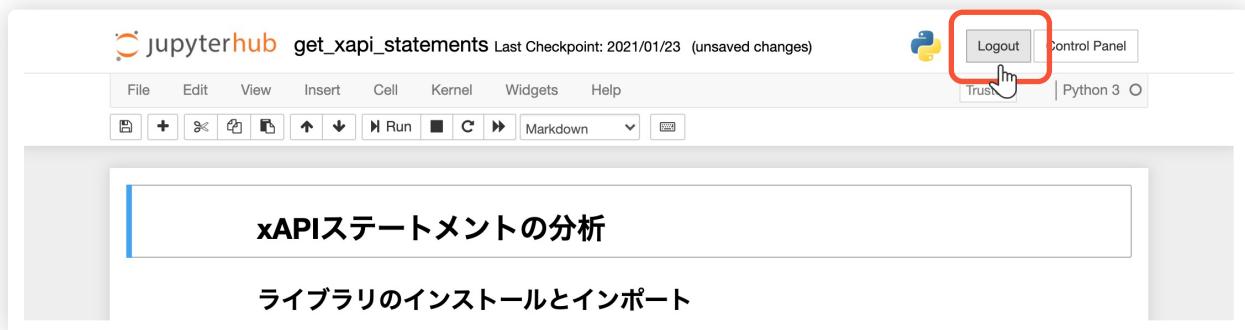
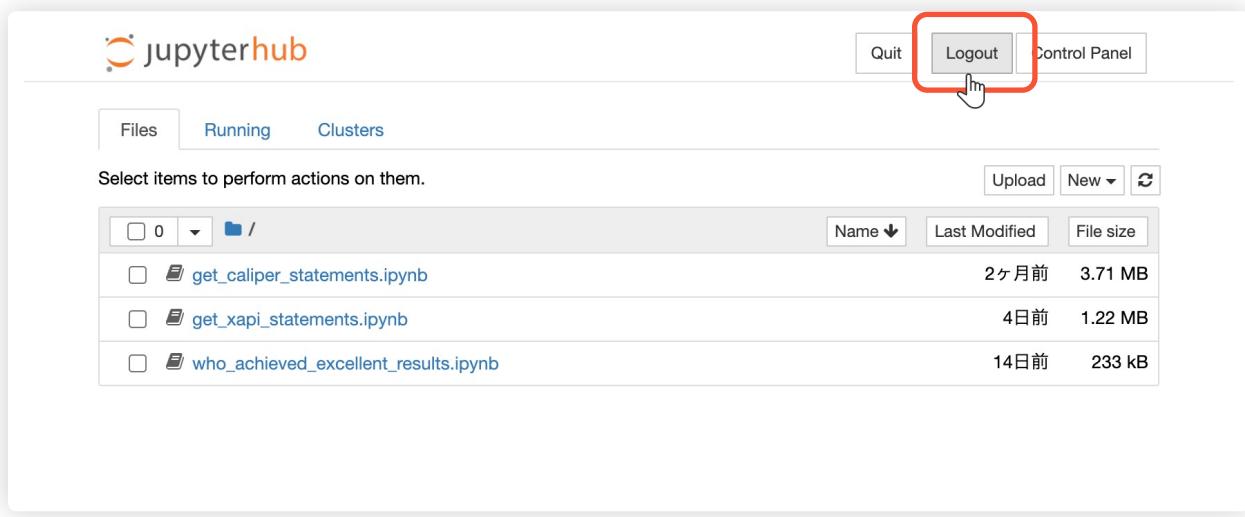
一覧にサンプルとして提供する以下のノートブックが存在することを確認する。

#	ファイル名	概要
1	get_caliper_statements.ipynb	Caliperステートメントを取得する例
2	get_xapi_statements.ipynb	xAPIステートメントを取得する例
3	who_achieved_excellent_results.ipynb	xAPIステートメントの小テスト結果から成績優秀者を求める例

次▶ 1.1.2 JupyterHubからログアウトする

1.1.2 | JupyterHubからログアウトする

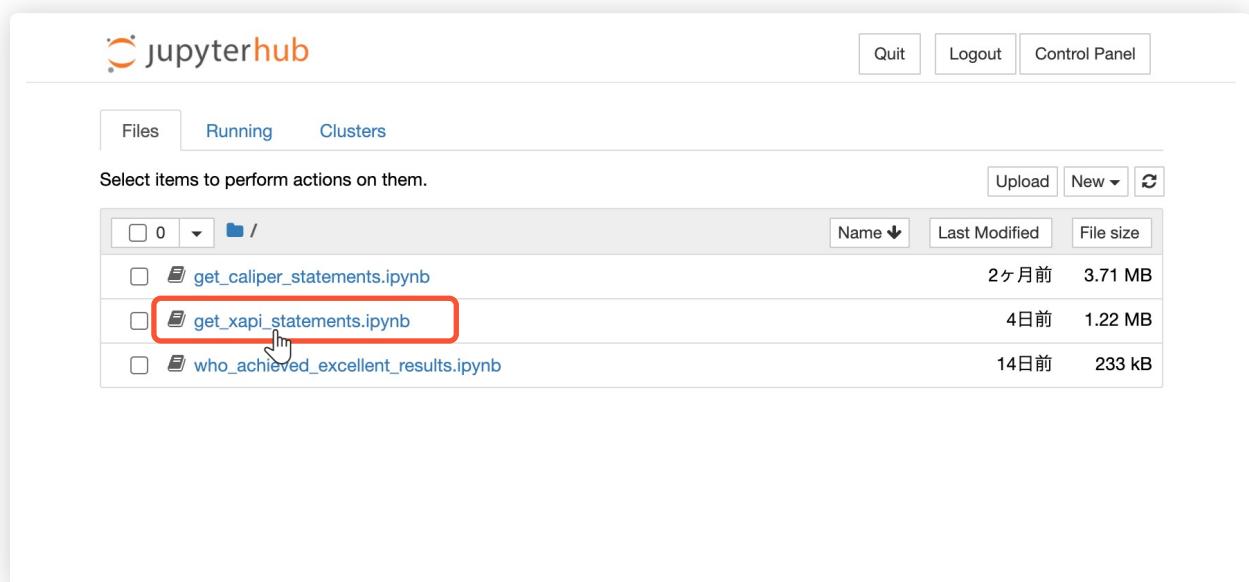
- ▶ 右上メニューの[Logout]ボタンをクリックしてログアウトする。メニュー位置は、ノートブック実行画面でも同様。ノートブック一覧ページへ遷移時は別タブが開かれるため、JupyterHubトップページのタブは予め閉じておく。
- ▶ ログアウトできずにノートブック一覧ページに遷移してしまう場合は、ブラウザのキャッシュを削除した上で[Logout]ボタンをクリックする。



次▶ 1.1.3 notebookを利用する

1.1.3 | notebookを利用する

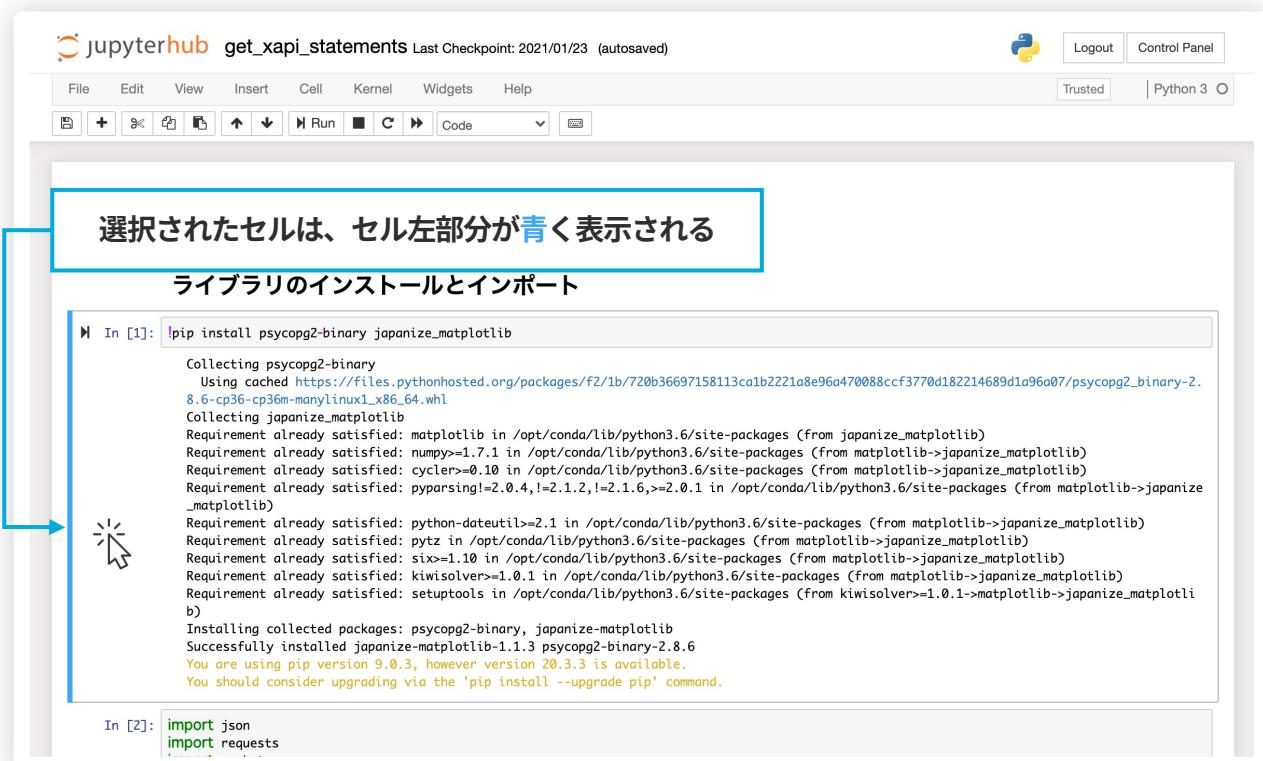
- ▶ ノートブック一覧から、利用したいノートブック名をクリックする。ここでは、`get_xapi_statements.ipynb`をクリックしたとして説明をする。



▶ 次ページへ

1. 基本操作 > 1.1 JupyterHubの操作 > 1.1.3 notebookを利用する

- ▶ ノートブックの実行画面に遷移する。この画面上でコードの記述、実行が可能。なお、初回アクセス時にはデフォルトのコードが表示される。
- ▶ コードはセル(Cell)単位で分割されており、セルごとに実行していく。セルを選択する場合はクリックする。

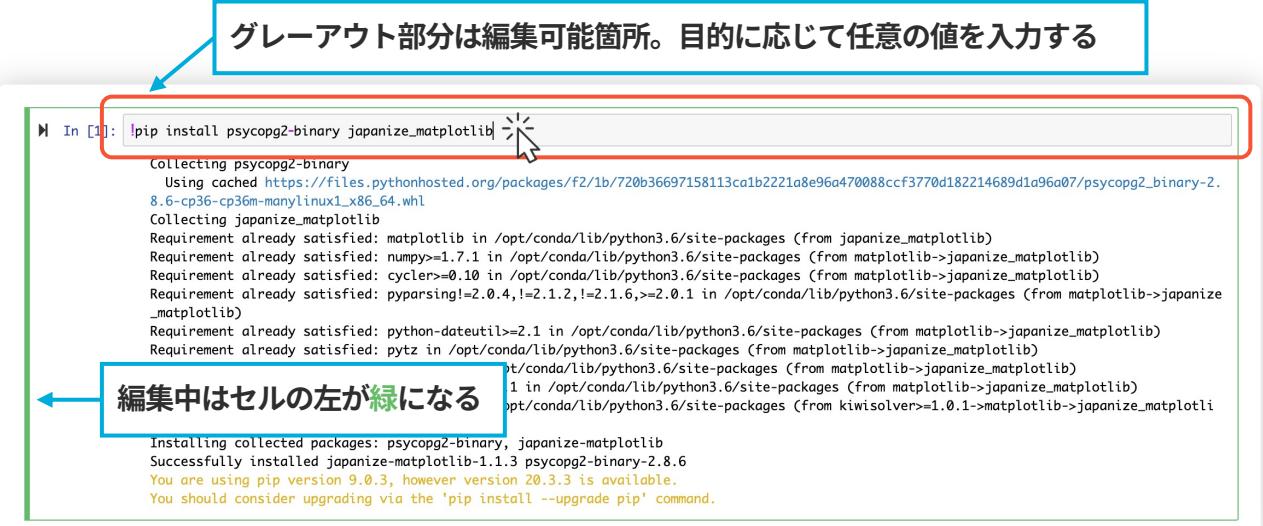


選択されたセルは、セル左部分が青く表示される

ライブラリのインストールとインポート

```
In [1]: pip install psycopg2-binary japanize_matplotlib
Collecting psycopg2-binary
  Using cached https://files.pythonhosted.org/packages/f2/1b/720b36697158113ca1b2221a8e96a470088ccf3770d182214689d1a96a07 psycopg2_binary-2.8.6-cp36-cp36m-manylinux1_x86_64.whl
Collecting japanize_matplotlib
  Requirement already satisfied: matplotlib in /opt/conda/lib/python3.6/site-packages (from japanize_matplotlib)
  Requirement already satisfied: numpy>=1.7.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
  Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
  Requirement already satisfied: pyparsing!=2.0.4,!>=2.1.2,!>=2.1.6,>=2.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
  Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
  Requirement already satisfied: pytz in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
  Requirement already satisfied: six>=1.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
  Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
  Requirement already satisfied: setuptools in /opt/conda/lib/python3.6/site-packages (from kiwisolver>=1.0.1->matplotlib->japanize_matplotlib)
Installing collected packages: psycopg2-binary, japanize_matplotlib
Successfully installed japanize-matplotlib-1.1.3 psycopg2-binary-2.8.6
You are using pip version 9.0.3, however version 20.3.3 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

```
In [2]: import json
import requests
```



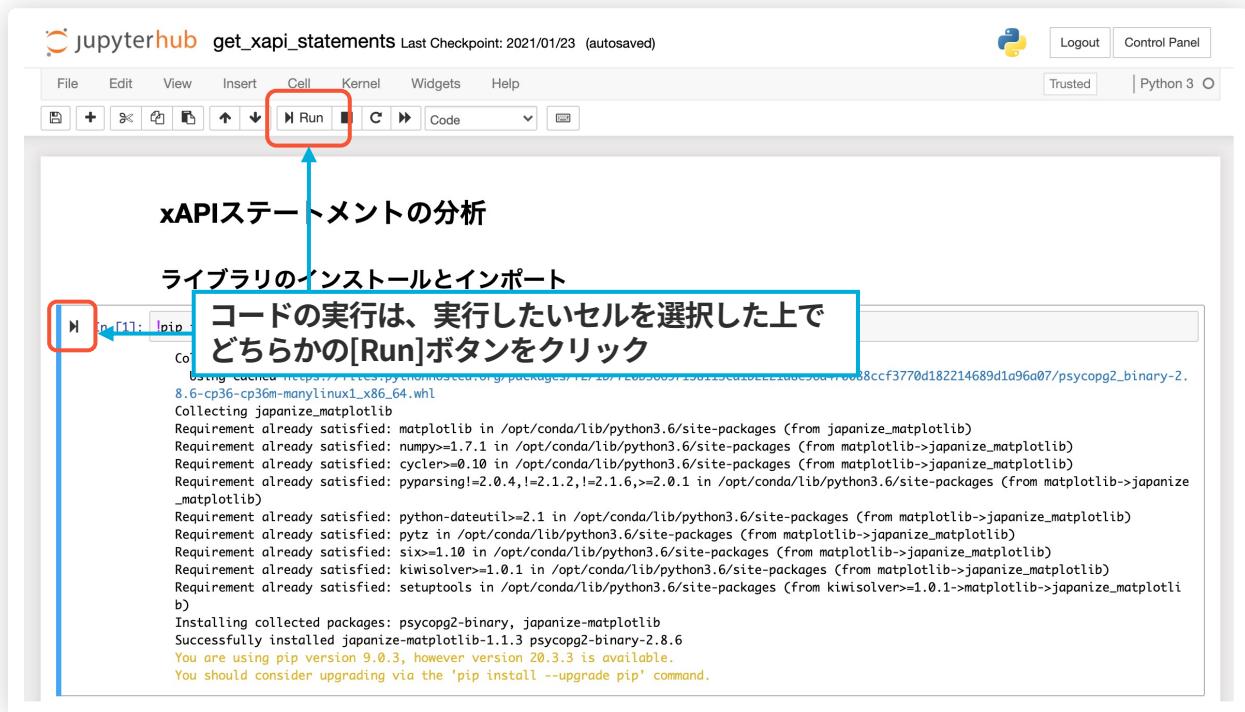
グレーアウト部分は編集可能箇所。目的に応じて任意の値を入力する

編集中はセルの左が緑になる

```
In [1]: pip install psycopg2-binary japanize_matplotlib
```

```
In [2]: import json
import requests
```

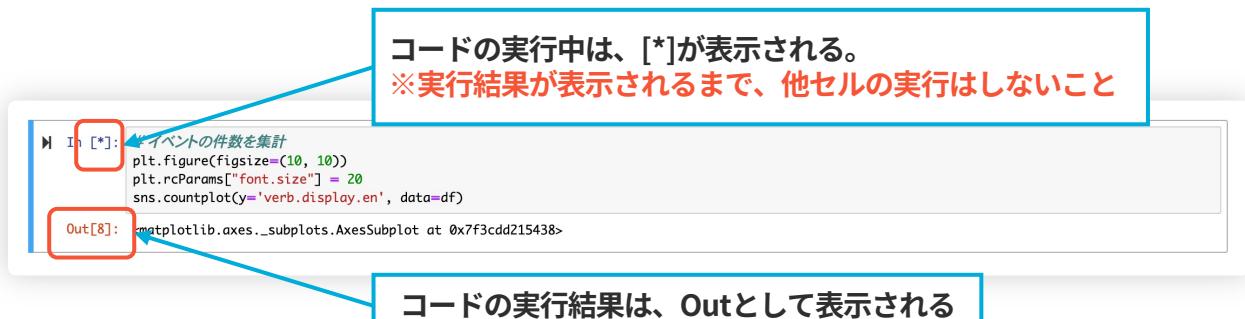
1. 基本操作 > 1.1 JupyterHubの操作 > 1.1.3 notebookを利用する



The screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the toolbar is a menu bar with File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Run, Trusted, and Python 3. A red box highlights the 'Run' button in the toolbar. The main area contains a code cell with the title 'xAPIステートメントの分析' and the subtitle 'ライブラリのインストールとインポート'. A red box highlights the 'In [1]:' label. Another red box highlights the 'Run' button again. A callout box with a blue border contains the text: 'コードの実行は、実行したいセルを選択した上でどちらかの[Run]ボタンをクリック'. The code cell contains the following Python code and output:

```
Using cached https://files.pythonhosted.org/packages/fe/10/fe00500f150115c010EE100C30d710688ccf3770d182214689d1a96a07/psycopg2_binary-2.8.6-cp36-cp36m-manylinux1_x86_64.whl
Collecting japanize_matplotlib
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.6/site-packages (from japanize_matplotlib)
Requirement already satisfied: numpy>=1.7.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: pyparsing!=2.0.4,!~2.1.2,!~2.1.6,>=2.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: pytz in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: six>=1.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.6/site-packages (from kiwisolver>=1.0.1->japanize_matplotlib)
b)
Installing collected packages: psycopg2-binary, japanize_matplotlib
Successfully installed japanize_matplotlib-1.1.3 psycopg2-binary-2.8.6
You are using pip version 9.0.3, however version 20.3.3 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

▶ コードは基本的に上から順に実行していく。



The screenshot shows a Jupyter Notebook interface. A red box highlights the 'In [1]:' label. Another red box highlights the 'Out[8]:' label. A callout box with a blue border contains the text: 'コードの実行中は、[*]が表示される。※実行結果が表示されるまで、他セルの実行はしないこと'. The code cell contains the following Python code and output:

```
In [1]: #イベントの件数を集計
plt.figure(figsize=(10, 10))
plt.rcParams["font.size"] = 20
sns.countplot(y='verb.display.en', data=df)

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3cdd215438>
```

A callout box with a blue border contains the text: 'コードの実行結果は、Outとして表示される'.



The screenshot shows a Jupyter Notebook interface. A red box highlights the 'In [10]:' label. Another red box highlights the 'Out[10]:' label. A callout box with a blue border contains the text: '[]内の数字はコードを実行した順序を示す'. The code cell contains the following Python code and output:

```
In [10]: #イベントの件数を集計
plt.figure(figsize=(10, 10))
plt.rcParams["font.size"] = 20
sns.countplot(y='verb.display.en', data=df_videojs)

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3cdd66cc50>
```

次▶ 1.2.1 Supersetへログインする

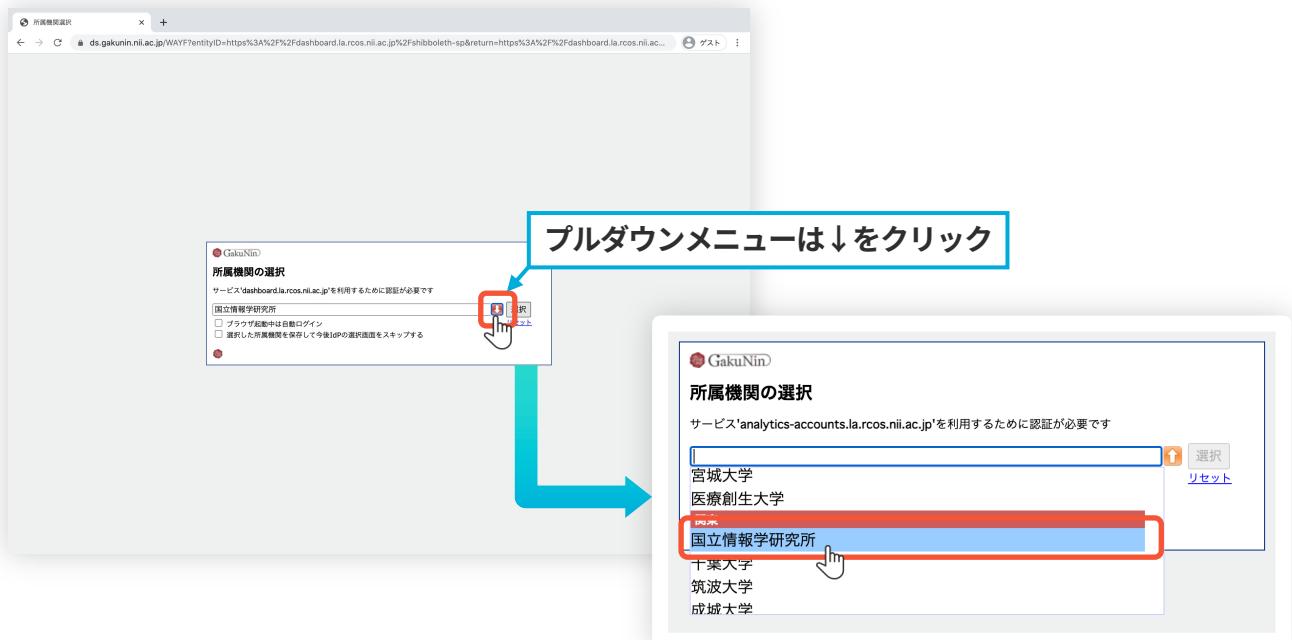
1. 基本操作 > 1.2 Supersetの操作 > 1.2.1 Supersetへログインする

1.2.1 | Supersetへログインする

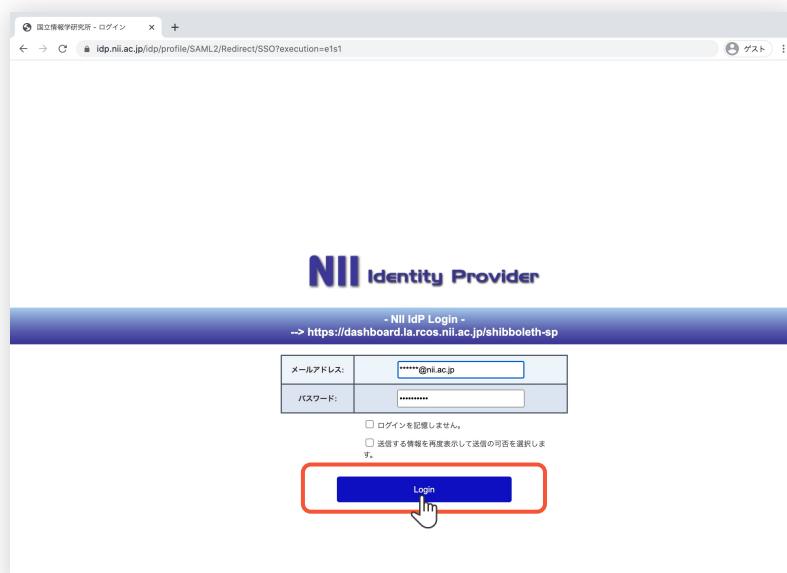
Supersetとは

GUIによる操作でデータの可視化、分析が可能なWEBアプリケーション

- ▶ <https://dashboard.la.rcos.nii.ac.jp>へアクセスし、自分が所属する機関名を選択する。

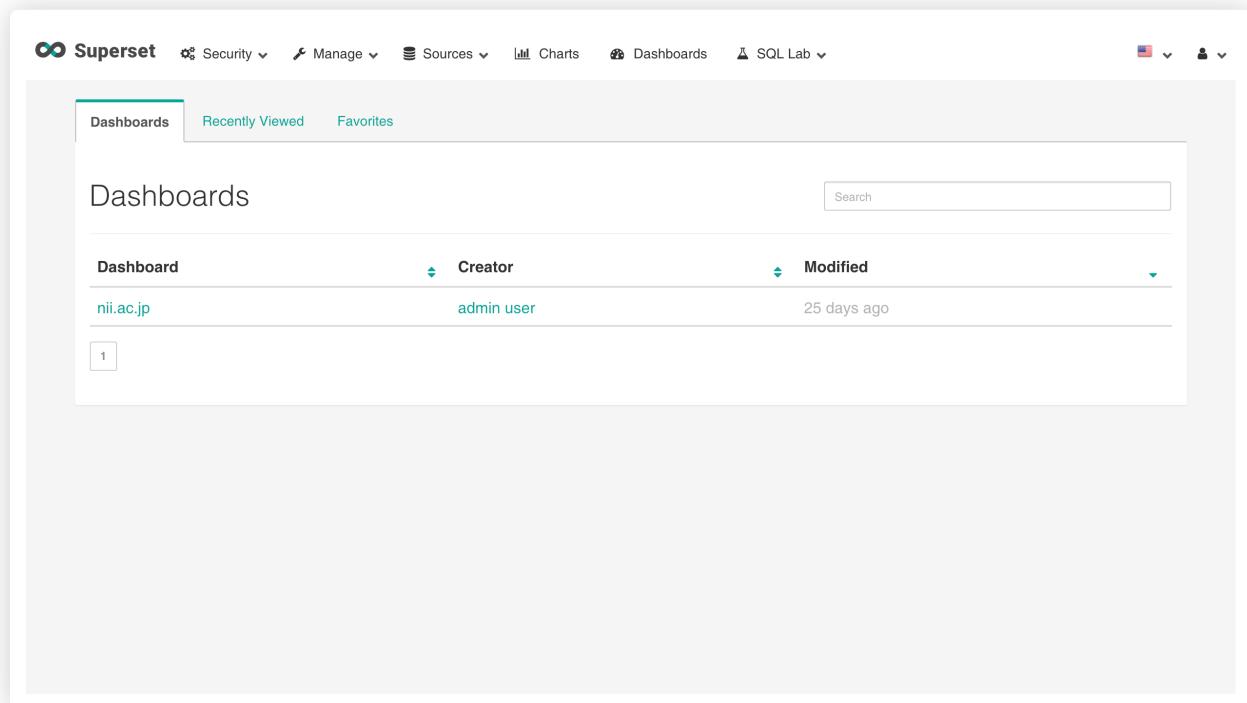


- ▶ 選択した所属機関のログインページに遷移するので、付与されているメールアドレスとパスワードを入力し[Login]ボタンをクリックする。



1. 基本操作 > 1.2 Supersetの操作 > 1.2.1 Supersetへログインする

- ▶ ログインが完了すると、ダッシュボード画面が表示される。ダッシュボード一覧には、所属している機関ドメイン名が表示される。

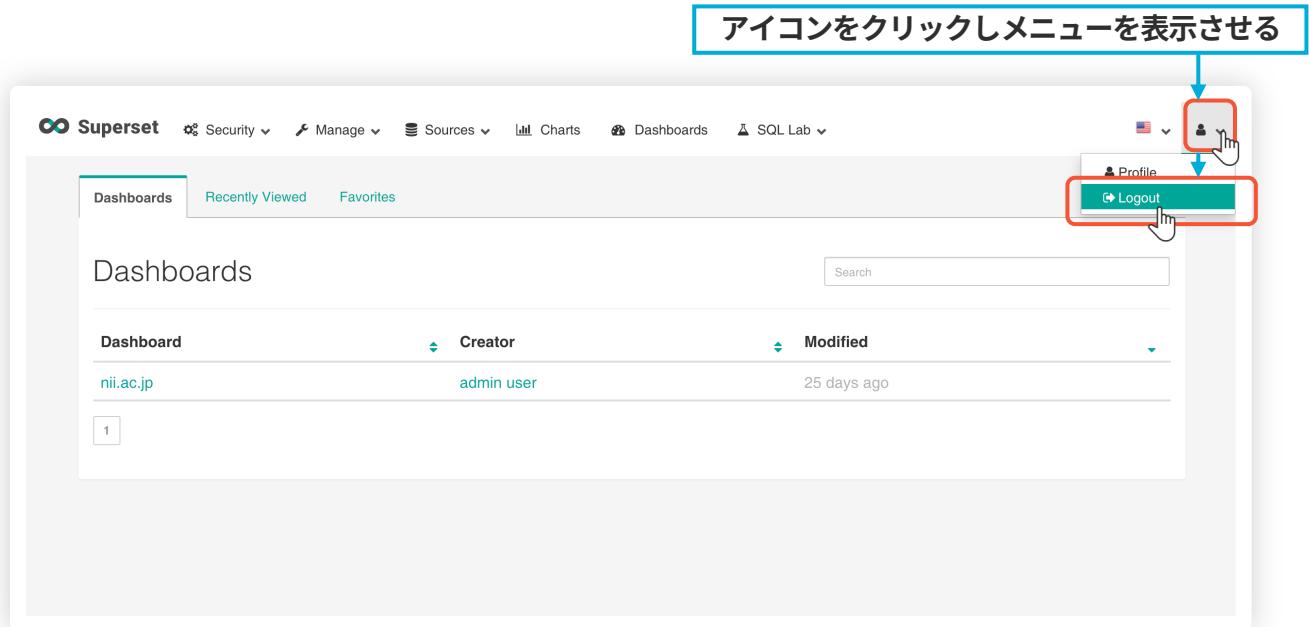


The screenshot shows the Superset application interface. At the top, there is a navigation bar with links for Security, Manage, Sources, Charts, Dashboards, and SQL Lab. On the far right of the navigation bar are icons for switching language (American English) and user profile. Below the navigation bar, there is a header with tabs: Dashboards (which is selected and highlighted in blue), Recently Viewed, and Favorites. To the right of these tabs is a search input field labeled "Search". The main content area is titled "Dashboards". It contains a table with one row of data. The columns are "Dashboard", "Creator", and "Modified". The data row shows "nii.ac.jp" under "Dashboard", "admin user" under "Creator", and "25 days ago" under "Modified". There is also a small thumbnail image of the dashboard, which appears to be a single card labeled "1".

次▶ 1.2.2 Supersetからログアウトする

1.2.2 | Supersetからログアウトする

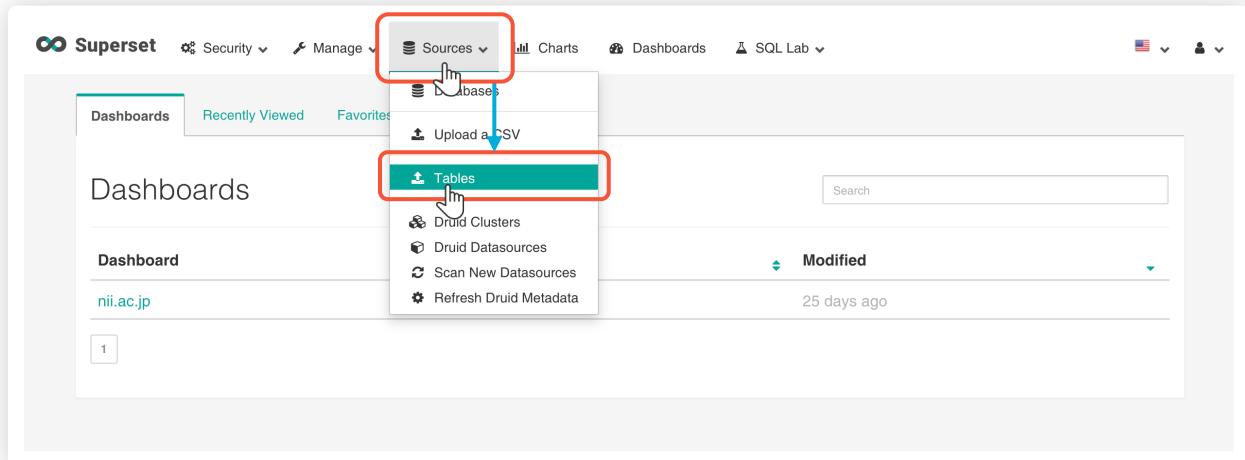
- ▶ 右上の👤アイコンをクリックし、メニューから[Logout]ボタンをクリックしてログアウトする。



次▶ 1.2.3 Supersetを利用する

1.2.3 | Supersetを利用する

- ▶ ダッシュボード画面上部メニューの[Sources]より、[Tables]をクリックする。



- ▶ 分析可能なデータの一覧が表示される。分析対象のテーブル名をクリックする。
ここでは、xapi_statements_nii_ac_jpをクリックしたとして説明する。

The screenshot shows the 'List Tables' page in Superset. The title is 'List Tables'. There is a header row with columns: Table, Database, Changed By, Modified, and Actions. Two rows of data are listed: 'xapi_statements_nii_ac_jp' (Database: learninglocker, Changed By: admin user, Modified: 25 days ago) and 'caliper_statements_nii_ac_jp' (Database: openlrw, Changed By: admin user, Modified: 25 days ago). The first row, 'xapi_statements_nii_ac_jp', is highlighted with a red box and a cursor icon pointing to it. At the bottom left, there is an 'Actions' dropdown menu. At the bottom right, it says 'Record Count: 2'.

Table	Database	Changed By	Modified	Actions
xapi_statements_nii_ac_jp	learninglocker	admin user	25 days ago	
caliper_statements_nii_ac_jp	openlrw	admin user	25 days ago	

1. 基本操作 > 1.2 Supersetの操作 > 1.2.3 Supersetを利用する

テーブルは一覧上部のFilterメニューより、検索と絞り込みが可能。

The screenshot shows the 'List Tables' interface. At the top, there are two filter sections: one for 'SQL Lab View' set to 'Not Equal to' and another for 'Database' set to 'jupyter'. A 'Search' button is highlighted with a red box and a cursor icon. A callout box with a blue border and arrow points to the 'Add Filter' button in the top right corner, which is also highlighted with a red box and a cursor icon. A modal window titled 'Database' is open, listing 'Schema', 'Table Name', 'Owner', and 'SQL Lab View'.

フィルターは複数設定が可能

[Add Filter]ボタンをクリックすると
絞り込みメニューが表示される

フィルターの設定後、[Search]ボタンを
クリックすると下部リストに反映される

テーブル一覧に目的のテーブルが存在しない場合は、以下の手順で追加する。

The screenshot shows the 'List Tables' page with a large blue L-shaped arrow pointing from the top section to the bottom section. In the bottom section, a 'New Record' form is displayed. It has fields for 'Database' (set to 'jupyter'), 'Schema' (dropdown menu), and 'Table Name' (input field containing 'xapi_statements_nii_ac_jp'). A callout box highlights the 'Database' selection. Another callout box highlights the 'Table Name' input field. A third callout box highlights the 'Save' button at the bottom left of the form.

List Tables 右の + ボタンから
テーブル作成ページへ遷移する

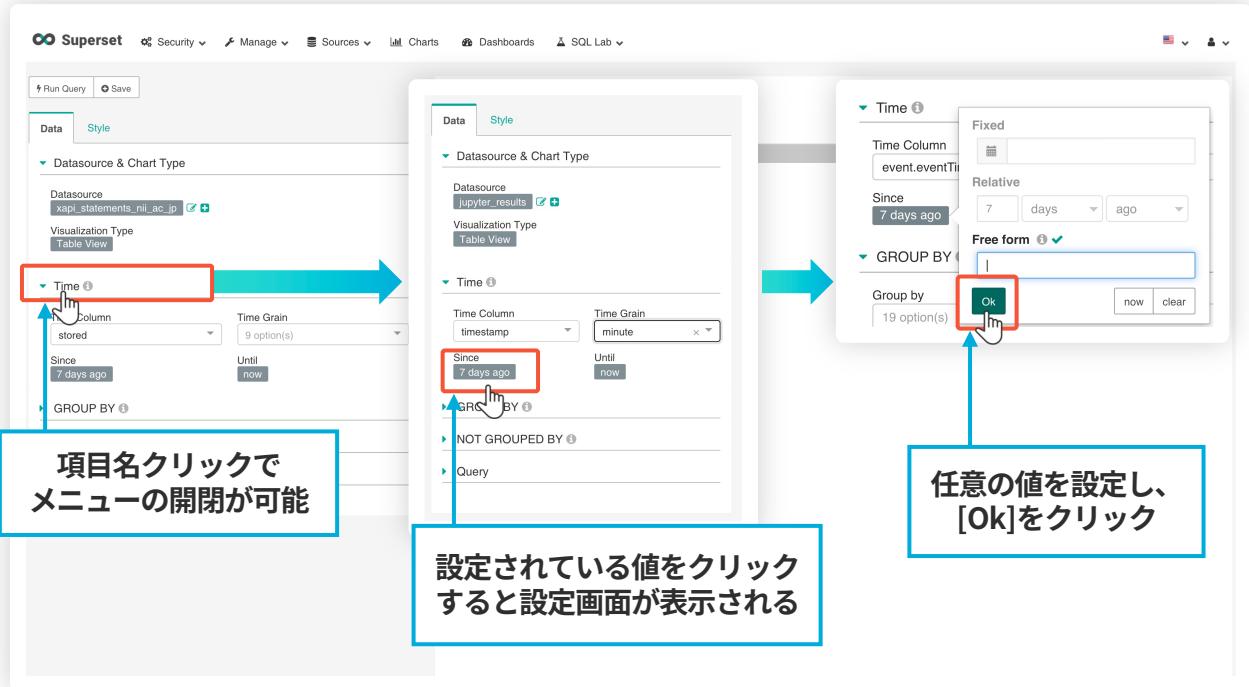
Databaseは jupyter を選択

Table NameにはJupyterノートブックで指定した
テーブル名を入力(この例ではxapi_statements_nii_ac_jp)

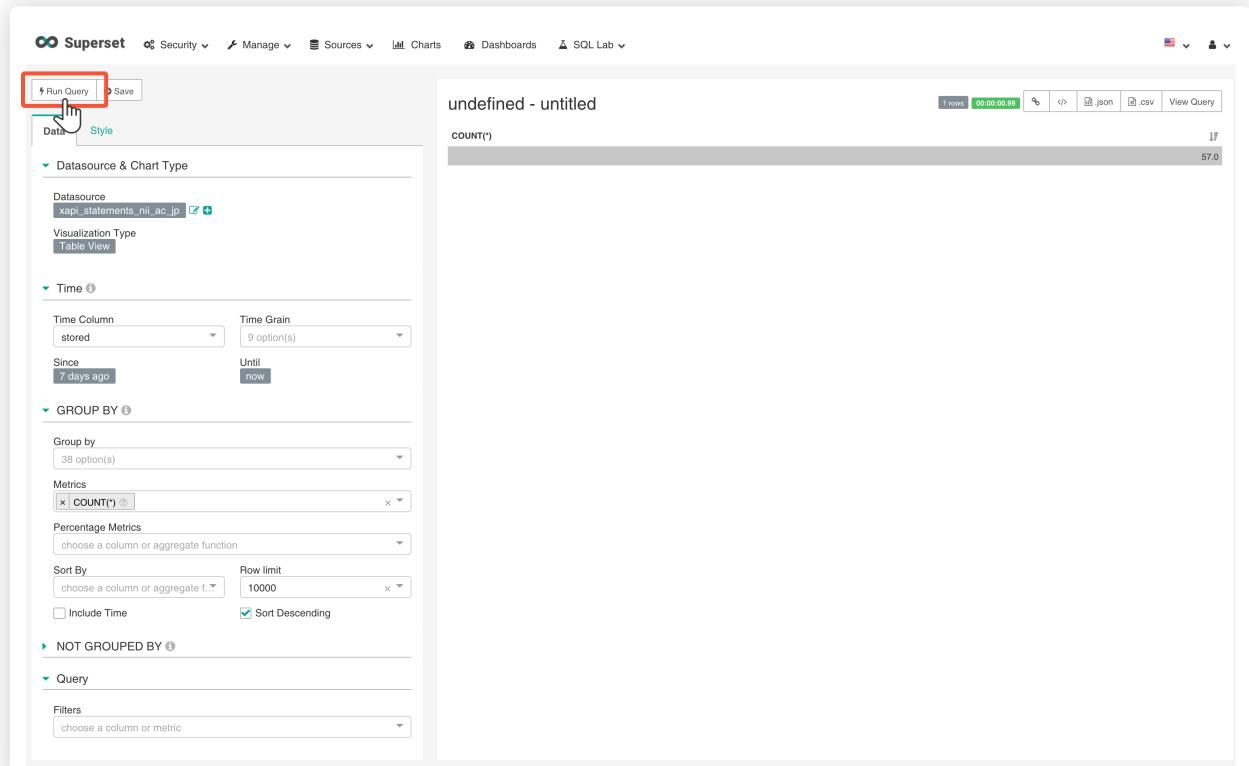
[Save]ボタンをクリックして追加

1. 基本操作 > 1.2 Supersetの操作 > 1.2.3 Supersetを利用する

- ▶ 左部メニューより、パラメータをそれぞれ設定する。



- ▶ 設定完了後に、[Run Query]ボタンをクリックすると、右部分に結果が表示される。



次▶ 2. 実践操作

2. 実践操作

2.1 動画の視聴回数を調べる 17p

2.1.1 JupyterHubの操作 18p

2.1.2 Supersetの操作 20p

2.2 小テストの結果から成績優秀者を特定する 21p

2.2.1 JupyterHubの操作 22p

2.2.2 Supersetの操作 24p

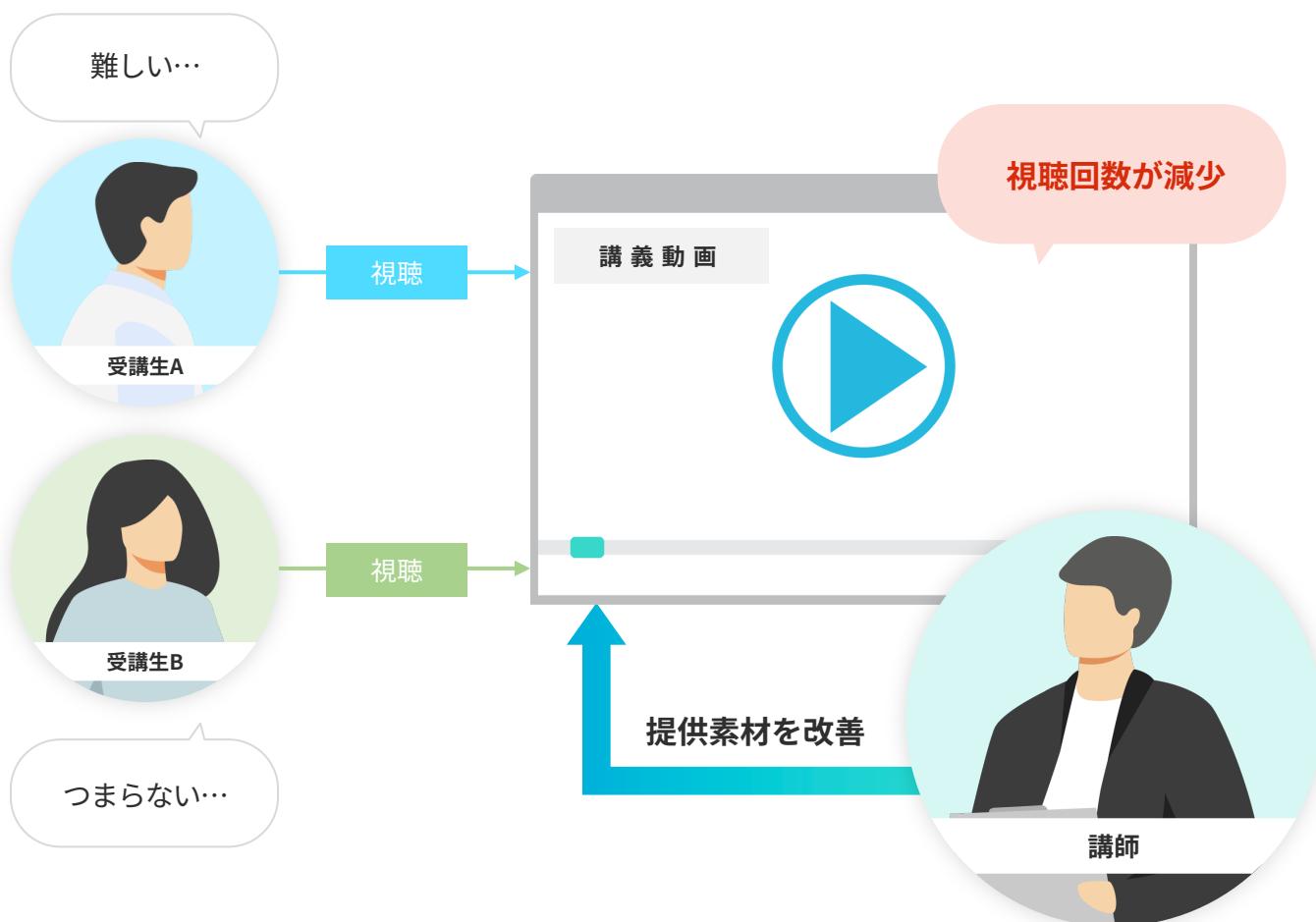
2.3 小テストの結果と動画の視聴履歴から退学者を予測する 27p

2.3.1 JupyterHubの操作 28p

【例1】動画の視聴回数の推移を調べる

講義動画の視聴回数の推移を把握することで、視聴回数が減少した際に、提供教材の改善を実施する等の対策を行うことができる。

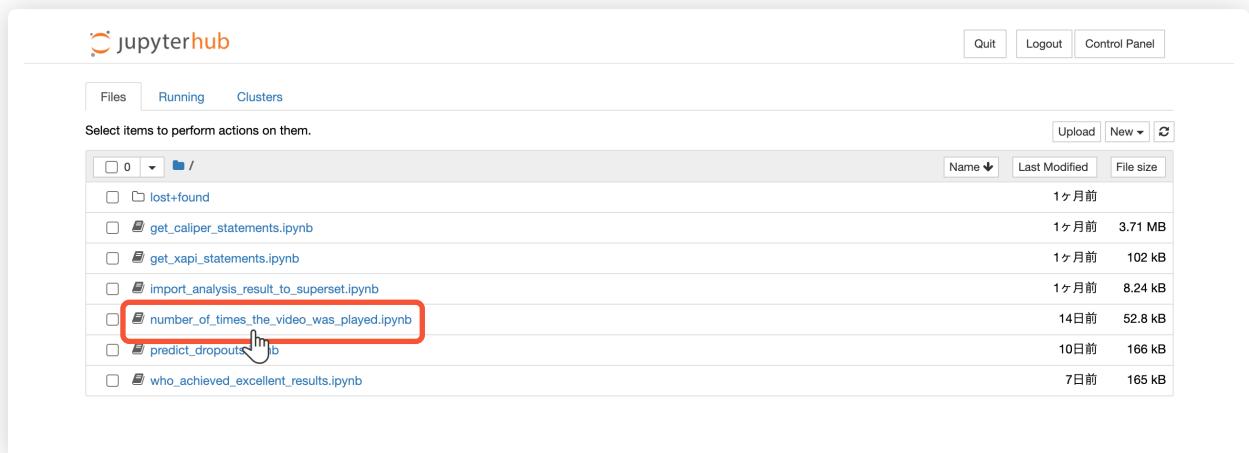
本シナリオでは、動画の視聴回数の推移を分析する手法を例示する。



▶次ページへ

2.1.1 | JupyterHubの操作

- ▶ 手順1.1の通りJupyterHubへログインし、ノートブックの一覧から number_of_times_the_video_was_played.ipynb を選択する。
- ▶ 本ノートブックは、動画の視聴データをもとに日毎の動画の視聴数を求め、実行結果を Supersetに登録する。



主な処理内容は以下の通り。(詳細はノートブックを参照すること)

- 学習ログ(xAPIステートメント)を取得
- コース名を取得
- 指定したコース内の動画名を取得
- 指定した動画の再生ログを取得
- 期間を設定し、日毎のイベント発生数を表示

▶ 次ページへ

2. 実践操作 > 2.1 動画の視聴回数を調べる > 2.1.1 JupyterHubの操作

- ▶ 上部セルよりコードを実行する。任意の値を指定する必要がある場合は都度入力する。今回は、コースIDを'31'、動画名を'RDM01/RDM01week0103.mp4'としている。

The screenshot shows a JupyterHub interface with the following details:

- Header:** jupyterhub number_of_times_the_video_was_played Last Checkpoint: 2021/01/25 (autosaved)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Logout, Control Panel, Trusted, Python 3.
- In [6]:** course_id = '31' (highlighted with a red box)
- In [7]:** video_names = []
for _, row in df.iterrows():
 if row['verb.display.en'] == 'firstplay' and
 row['object.id'] in object_names[course_id]
 row['object.definition.name.en'] not in video_names:
 video_names.append(row['object.definition.name.en'])
video_names
- Out[7]:** ['RDM01/RDM01week0101.mp4', 'RDM01/RDM01week0103.mp4']
- In [14]:** video_name = 'RDM01/RDM01week0103.mp4' (highlighted with a red box)

A blue arrow points from the text "変更する場合は、調べたい対象に応じた値を入力する" (When changing, enter the value corresponding to the target) to the input field In [6]. Another blue arrow points from the same text to the input field In [14].

- ▶ 『jupyterデータベースに処理結果を保存』セルが正常に実行されたことを確認し、JupyterHubでの操作を完了させる。

The screenshot shows a JupyterHub interface with the following details:

- In [19]:** pip install psycopg2-binary
from sqlalchemy import create_engine
from sqlalchemy.types import TIMESTAMP

所属機関に割り当てられたユーザ情報を設定する
username =
password =

dfq = dfq.drop('context.contextActivities.category', axis=1)
engine = create_engine('postgresql://'+ username + ':' + password +'
tablename = 'number_of_times_the_video_was_played'
dfq.to_sql(tablename, engine, if_exists='replace', dtype={'timestamp': TIMESTAMP})
- Output:** Requirement already satisfied: psycopg2-binary in /opt/conda/lib/python3.6/site-packages
You are using pip version 9.0.3, however version 21.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.

A blue arrow points from the text "データベースへの保存処理セルを必ず実行" (Always execute the save processing cell to the database) to the In [19] cell. Another blue arrow points from the text "Supersetでのテーブル名はこの部分を参照のこと" (Refer to this part for the table name in Superset) to the line of code "tablename = 'number_of_times_the_video_was_played'".

2. 実践操作 > 2.1 動画の視聴回数を調べる > 2.1.2 Supersetの操作

2.1.2 | Supersetの操作

- ▶ 手順1.2の通りSupersetへログインし、JupyterHubの処理結果テーブルを選択する。

The screenshot shows the Superset interface with the title 'List Tables'. There is a search bar and a filter dropdown set to 'Not Equal to'. Below is a table with three rows. The first row, 'number_of_times_the_video_was_played', is highlighted with a red box and has a cursor pointing at it. The other two rows are 'jupyter' and 'jupyter_results'.

- ▶ 左部メニューより各項目の値を設定し、[Run Query]をクリックしてデータを可視化する。

期間を指定して日毎のイベント発生数を表示

The screenshot shows the Superset configuration interface for a 'Time Series - Bar Chart'. On the left, there's a sidebar with 'Run Query' and 'Save' buttons, and sections for 'Datasource & Chart Type' (selected 'number_of_times_the_video_was_played'), 'Time' (Time Column: 'timestamp', Time Grain: '9 option(s)', Since: '100 days ago', Until: 'now'), and 'Query' (Metrics: 'COUNT(*)'). The main area shows a bar chart titled 'undefined - untitled' with three bars at y=1.000. A legend indicates 'COUNT(*)'. A large blue L-shaped arrow points from the sidebar towards the chart area. Below the chart is a detailed table of settings:

タブ	大項目	小項目	設定値
Data	Visualization Type	Time Series - Bar Chart	
	Time	Time Column	timestamp
		Since	任意の日付
		Until	任意の日付
	Query	Metrics	COUNT(*)
	Advanced Analytics	Resample How	sum
		Resample Rule	1D
Style	X Axis	X Axis Format	%Y-%m-%d
	Y Axis	Y Axis Format	,.3f

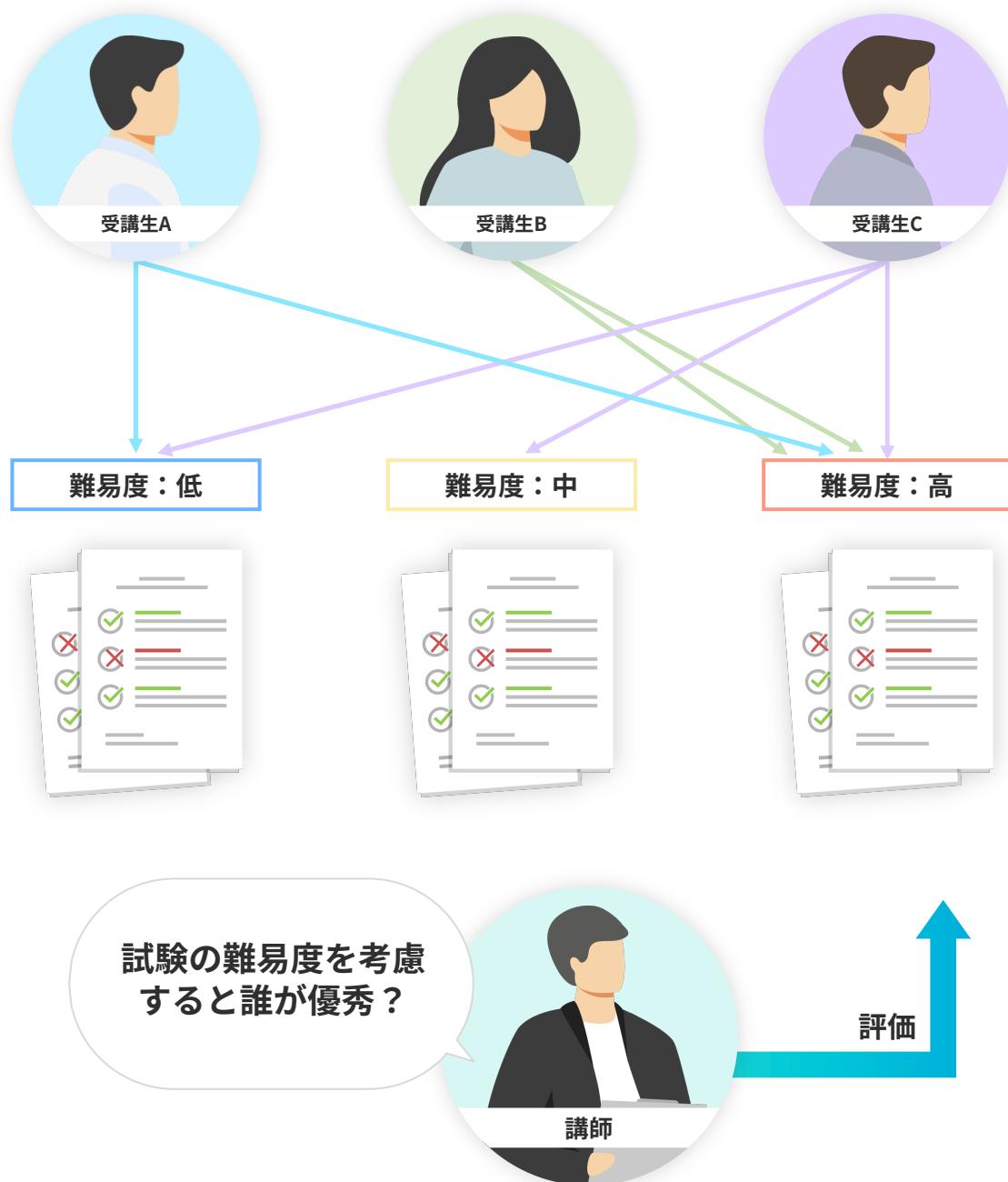
次▶ 2.2 小テストの結果から成績優秀者を特定する

【例2】小テストの結果から成績優秀者を特定する

学認LMSでは、講師は各講義の中で小テストを実施、採点することができる。

講師が受講生の中から成績優秀者を特定する際に、小テストはそれぞれ難易度が異なるため、単純な採点の合計は受講生の優劣を決定するためには適切でない恐れがある。

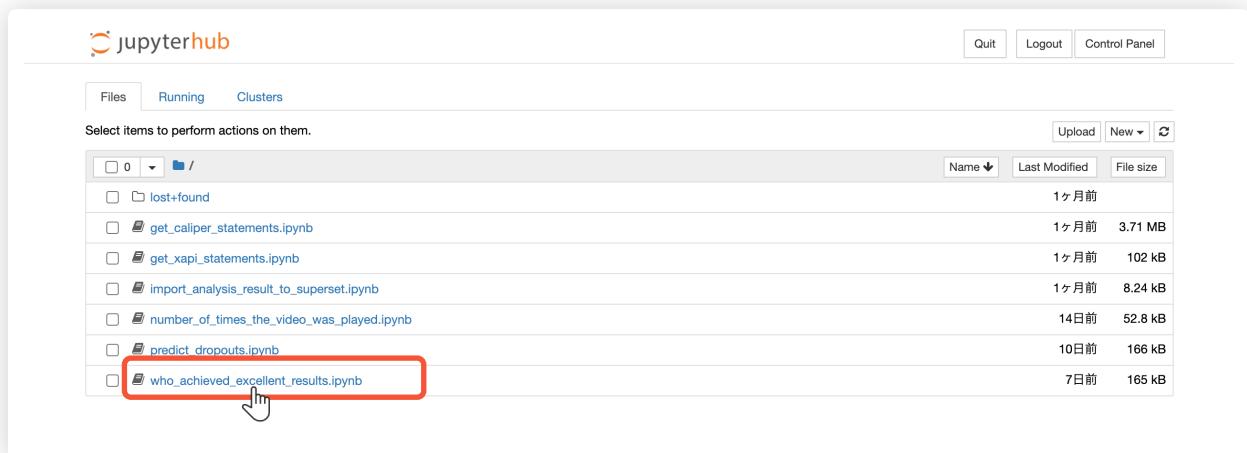
したがって、本シナリオでは、各小テストの難易度を考慮して成績優秀者を特定する分析方法を例示する。



▶ 次ページへ

2.2.1 | JupyterHubの操作

- ▶ 手順1.1の通りJupyterHubへログインし、ノートブックの一覧から who_achieved_excellent_results.ipynb を選択する。
- ▶ 本ノートブックは小テスト結果から成績優秀者を求め、実行結果をSupersetに登録する。



主な処理内容は以下の通り。(詳細はノートブックを参照すること)

- 学習ログ(xAPIステートメント)を取得
- 小テスト結果(完了済かつ及第点)のみを取得
- コースの日本語名列を追加
- テスト受験時間 (result.duration) をISO8601形式から数値形式（分単位）に変換
- テスト結果(result.score)からテストの正答率を求めてA～Dランク評価を追加
- テストごとに平均、標準偏差、分散、最小、最大を算出して追加
- 標準得点を (取得した得点 - 平均) / 標準偏差 でテストごとに算出
- テストごとの順位を追加
- 小テストごとの1位を一覧化
- 特定のコースを対象にユーザ単位に取得したグレードの数をカウント(※)
- 特定のコースを対象に標準得点の合計を表示(※)
- Supersetに加工した学習ログを登録

2. 実践操作 > 2.2 小テストの結果から成績優秀者を特定する > 2.2.1 JupyterHubの操作

- ▶ 上部セルよりコードを実行する。任意の値を指定する必要がある場合は都度入力する。

In [1]: `!pip install psycopg2-binary japanize_matplotlib`

Requirement already satisfied: psycopg2-binary in /opt/conda/lib/python3.6/site-packages
Requirement already satisfied: japanize_matplotlib in /opt/conda/lib/python3.6/site-packages
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.6/site-packages (from japanize_matplotlib)
Requirement already satisfied: numpy>=1.7.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: pytz in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: six>=1.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.6/site-packages (from kiwisolver>=1.0.1->matplotlib)
Requirement already satisfied: _matplotlib
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: pytz in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: six>=1.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.6/site-packages (from kiwisolver>=1.0.1->matplotlib)
b)
You are using pip version 9.0.3, however version 21.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.

In [2]: `import json
import requests
import socket

import japanize_matplotlib
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import psycopg2
import seaborn as sns
from sqlalchemy import create_engine`

In [4]: `# ステートメントをAPIサーバから取得(以下はxAPIステートメント取得の例)
所属機関に用意されたLearning Lockerのクライアント情報を設定
user = ''
passwd = ''`

In [5]: `## 5000件ずつステートメントを取得する(APIの詳細:https://docs.learninglocker.net/http-connection/)
url = 'https://lrs.lrcos.nii.ac.jp/api/connection/statement?first=5000'
Learning LockerのAGGREGATION APIでxAPIステートメントを取得`

- ▶ コードの実行結果を保存し、JupyterHubでの操作を完了させる。

In [24]: `# ダッシュボードサーバのjupyterデータベースに処理結果を保存
所属機間に割り当てられたユーザ情報を設定すること
username = '*****'
password = '*****'

engine = create_engine('postgresql://'+username+':'+password+'@'+hostname+'/' + tablename +'
tablename = 'who_achieved_excellent_results'

必要に応じてdtypeでカラムの属性を指定する
from sqlalchemy.types import TIMESTAMP
dfq.to_sql(tablename, engine, if_exists='replace', dtype={'timestamp':TIMESTAMP})`

データベースへの保存処理セルを必ず実行する

Supersetでのテーブル名はこの部分を参照のこと

2.2.2 | Supersetの操作

- ▶ 手順1.2の通りSupersetへログインし、JupyterHubの処理結果テーブルを選択する。

Table	Database	Changed By	Modified
who_achieved_excellent_results	jupyter	admin user	now
number_of_time_the_video_was_played	jupyter	admin user	7 days ago
jupyter_results	jupyter	admin user	10 days ago

- ▶ 左部メニューより各項目の値を設定し、[Run Query]をクリックしてデータを可視化する。

小テストごとの1位を一覧化

各設定項目のパラメータ設定例は以下の通り。

course	object.id	actor.account.name
	https://ms.nii.ac.jp/mod/quiz/view.php?id=250	337
	https://ms.nii.ac.jp/mod/quiz/view.php?id=244	304
	https://ms.nii.ac.jp/mod/quiz/view.php?id=245	304
	https://ms.nii.ac.jp/mod/quiz/view.php?id=245	304
	https://ms.nii.ac.jp/mod/quiz/view.php?id=671	488
	https://ms.nii.ac.jp/mod/quiz/view.php?id=671	503
	https://ms.nii.ac.jp/mod/quiz/view.php?id=590	467
	https://ms.nii.ac.jp/mod/quiz/view.php?id=590	467
	https://ms.nii.ac.jp/mod/quiz/view.php?id=590	477
	https://ms.nii.ac.jp/mod/quiz/view.php?id=590	509
	https://ms.nii.ac.jp/mod/quiz/view.php?id=590	437
	https://ms.nii.ac.jp/mod/quiz/view.php?id=590	402
	https://ms.nii.ac.jp/mod/quiz/view.php?id=590	405
	https://ms.nii.ac.jp/mod/quiz/view.php?id=590	346
	https://ms.nii.ac.jp/mod/quiz/view.php?id=595	402
	https://ms.nii.ac.jp/mod/quiz/view.php?id=595	405

大項目	小項目	設定値
Visualization Type		Table View
Time	Time Column	timestamp
	Since	∞
	Until	now
NOT GROUPED BY	Columns	course
	object.id	object.id
	actor.account.name	actor.account.name
	Ordering	course[asc]
	object.id[asc]	object.id[asc]
Query	Filters	rank = 1

2. 実践操作 > 2.2 小テストの結果から成績優秀者を特定する > 2.2.2 Supersetの操作

特定コースで各ユーザが取得した成績評価（A～D）の数を可視化

各設定項目のパラメータ設定例は以下の通り。

Superset Sources Charts Dashboards

Run Query Save

Data Style

Datasource & Chart Type

Datasource my_table

Visualization Type Distribution - Bar Chart

Time

Time Column timestamp

Time Grain 9 option(s)

Since ∞

Until now

Query

Metrics COUNT(*)

Filters course like <コース名>

Series actor.account.name

Breakdowns grade

Row limit 10000

Contribution

undefined - untitled

220 rows 00:00:00.39

json csv View Query

大項目	小項目	設定値
Visualization Type		Distribution - Bar Chart
Time	Time Column	timestamp
	Since	∞
	Until	now
Query	Metrics	COUNT(*)
	Filters	course like <コース名>
	Series	actor.account.name
	Breakdowns	grade

特定コース内で標準得点の合計値をユーザ毎に可視化

各設定項目のパラメータ設定例は以下の通り。

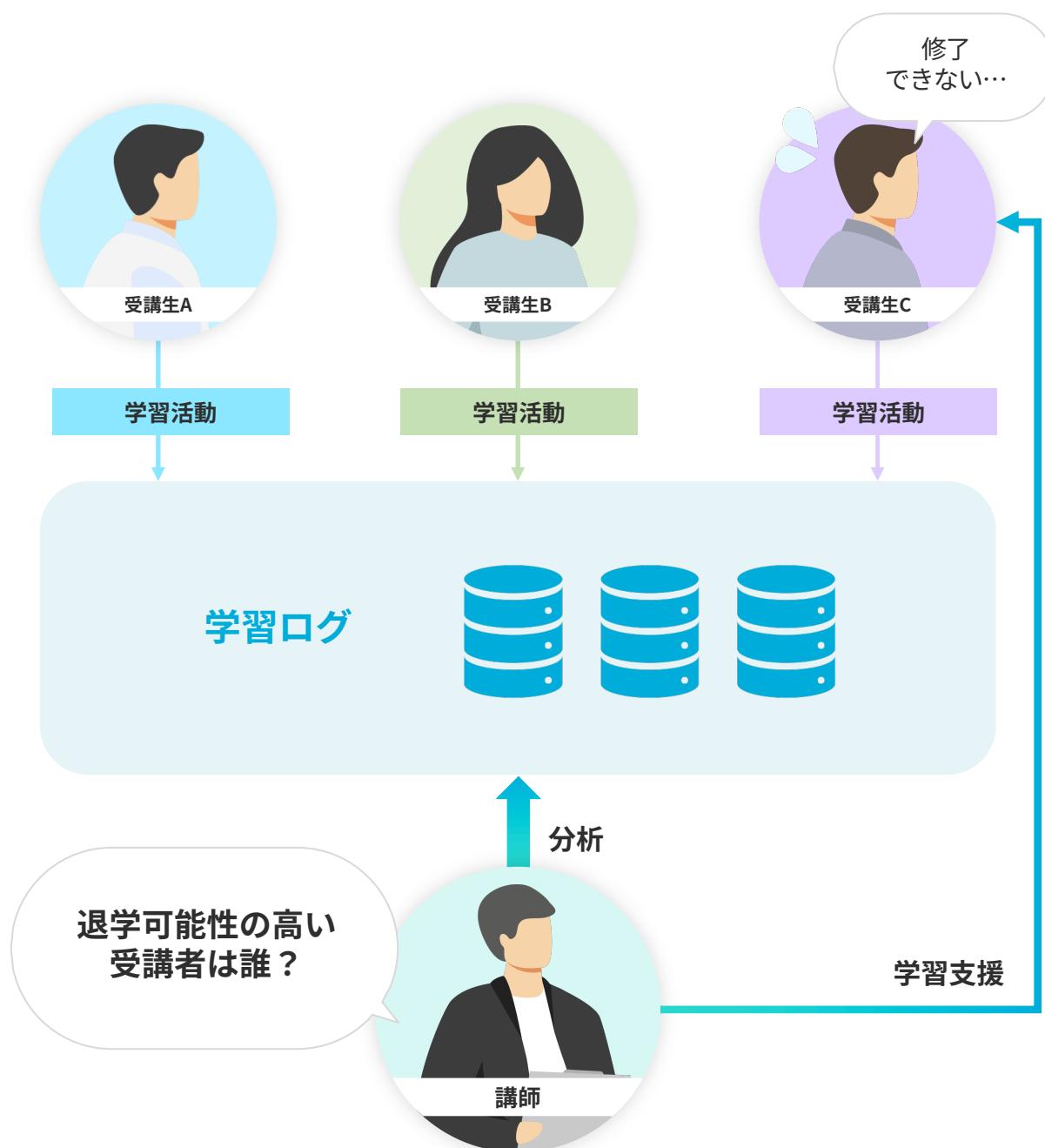
The screenshot shows the Superset interface with a bar chart titled "undefined - untitled". The chart displays user scores (z_scores) for different users. A specific user entry is highlighted with a black box, showing a value of 8.91. The interface includes a sidebar for "Datasource & Chart Type" (DataSource: my_table, Visualization Type: Distribution - Bar Chart), "Time" (Time Column: timestamp, Since: infinity, Until: now), and "Query" (Metrics: SUM(z_score), Filters: choose a column or metric, Series: actor.account.name, Breakdowns: course, Row limit: 10000). The chart has a y-axis ranging from -10.0 to 5.00.

大項目	小項目	設定値
Visualization Type		Distribution - Bar Chart
Time	Time Column	timestamp
	Since	∞
	Until	now
Query	Metrics	SUM(z_score)
	Breakdowns	course

次▶ 2.3 小テストの結果と動画の視聴履歴から退学者を予測する

【例3】小テストの結果と動画の視聴履歴から退学者を予測する

オンライン講座では、講座を最後まで修了できずに退学する受講者が多く存在する。退学可能性の高い受講者を推定して学習支援を行うことは、退学率の改善につながる。
本シナリオでは、小テストの結果と動画の視聴履歴から退学者を予測する方法を例示する。



▶ 次ページへ

2.3.1 | JupyterHubの操作

- ▶ 手順1.1の通りJupyterHubへログインし、ノートブックの一覧から predict_dropouts.ipynb を選択する。
- ▶ 本ノートブックは、小テストの結果と動画の視聴データをもとに退学可能性の高い受講生を予測する。

The screenshot shows the JupyterHub interface with a list of notebooks. The notebook 'predict_dropouts.ipynb' is highlighted with a red box. The interface includes a header with 'jupyterhub' logo, 'Quit', 'Logout', and 'Control Panel' buttons. Below the header are tabs for 'Files', 'Running', and 'Clusters'. A search bar says 'Select items to perform actions on them.' There are buttons for 'Upload', 'New', and a refresh icon. The main area lists notebooks with columns for Name, Last Modified, and File size.

Name	Last Modified	File size
lost+found	1ヶ月前	
get_caliper_statements.ipynb	1ヶ月前	3.71 MB
get_xapi_statements.ipynb	1ヶ月前	102 kB
import_analysis_result_to_superset.ipynb	1ヶ月前	8.24 kB
number_of_times_the_video_was_played.ipynb	14日前	52.8 kB
predict_dropouts.ipynb	10日前	166 kB
who_achieved_relevant_results.ipynb	7日前	165 kB

主な処理内容は以下の通り。(詳細はノートブックを参照すること)

- 学習ログ(xAPIステートメント)を取得
- コース名を取得
- 指定コース内の小テストidを取得
- 指定コース内のすべての小テストを受験済みのユーザを特定
- エルボー法によるクラスタ数の決定
- 小テストの点数をもとにユーザをクラスタリング
- クラスタを可視化
- 指定コース内の動画名を取得
- ユーザ毎に動画の視聴回数と視聴間隔を取得
- ランダムフォレストによる予測モデルの作成と評価
- 予測対象ユーザを特定
- 予測対象ユーザに関する動画の視聴履歴を取得
- 予測モデルの適用と結果の表示

2. 実践操作 > 2.3 小テストの結果と動画の視聴履歴から退学者を予測する > 2.3.1 JupyterHubの操作

- ▶ 上部セルよりコードを実行する。任意の値を指定する必要がある場合は都度入力する。

The screenshot shows the JupyterHub interface with the title bar "jupyterhub predict_dropouts Last Checkpoint: 2021/01/25 (autosaved)". The toolbar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a Trusted Python 3 button. In the top cell (In [8]), the code `course_id = '10'` is written, with the value '10' highlighted and a red box around the entire cell. A blue arrow points from this cell to the explanatory text below. The bottom cell (In [9]) contains Python code for filtering a DataFrame `dfq` based on specific conditions and extracting `object.id` values into `test_obj_ids`. A blue box highlights the explanatory text "変更する場合は、調べたい対象に応じた値を入力する" (When changing, enter the value corresponding to the target object).

```
In [8]: course_id = '10'

In [9]: dfq = df[
    (df['object.definition.name.en'] == 'attempt_submitted') &
    (df['result.completion'] == True)
].copy()
dfq.head()

test_obj_ids = []
for _, row in dfq.iterrows():
    if row['object.id'] in object_names[course_id] and \
        row['object.id'] not in test_obj_ids:
        test_obj_ids.append(row['object.id'])

test_obj_ids
```

変更する場合は、調べたい対象に応じた値を入力する

- ▶ 最終セルを実行して予測モデルを適用し、予測結果を表示させる。

The screenshot shows the JupyterHub interface with the same title bar and toolbar. In the bottom cell (In [23]), the code applies a classifier `clf` to `pred_target_data` and creates a DataFrame `dfr` with columns 'actor.account.name' and 'cluster'. The resulting DataFrame is displayed in Out[23], which is highlighted with a red box. A blue arrow points from this output to the explanatory text below. The output data is as follows:

actor.account.name	cluster
0	0
1	11
2	12
3	13

予測結果はOutに出力される

```
In [23]: pred_result = clf.predict(pred_target_data)

result_data = []
for i in range(len(pred_result)):
    result_data.append([target_users[i], pred_result[i]])

dfr = pd.DataFrame(data=result_data, columns=['actor.account.name', 'cluster'])

Out[23]:
```

ラーニングアナリティクス基盤 ユーザマニュアル

1.0版, 2021年3月31日

国立情報学研究所