

# Introduction to Elyra: AI-centric extensions to JupyterLab



Edward Leardi  
Saishruthi Swaminathan  
Yiwen Li

# About us



IBM – Center for Open Source Data and AI Technologies (CODAIT)

---



Yiwen Li  
Data Scientist



Edward Leardi  
Data Scientist



Saishruthi Swaminathan  
Developer Advocate & Data Scientist

---



[Yiwen.Li@ibm.com](mailto:Yiwen.Li@ibm.com)



[github.com/yil532](https://github.com/yil532)



[linkedin.com/in/yiwenli](https://linkedin.com/in/yiwenli)

[edward@ibm.com](mailto:edward@ibm.com)

[github.com/edwardleardi](https://github.com/edwardleardi)

[linkedin.com/in/edwardleardi](https://linkedin.com/in/edwardleardi)

[saishruthi.tn@ibm.com](mailto:saishruthi.tn@ibm.com)

<https://github.com/SSaishruthi>

<https://www.linkedin.com/in/saishruthi-swaminathan/>

---

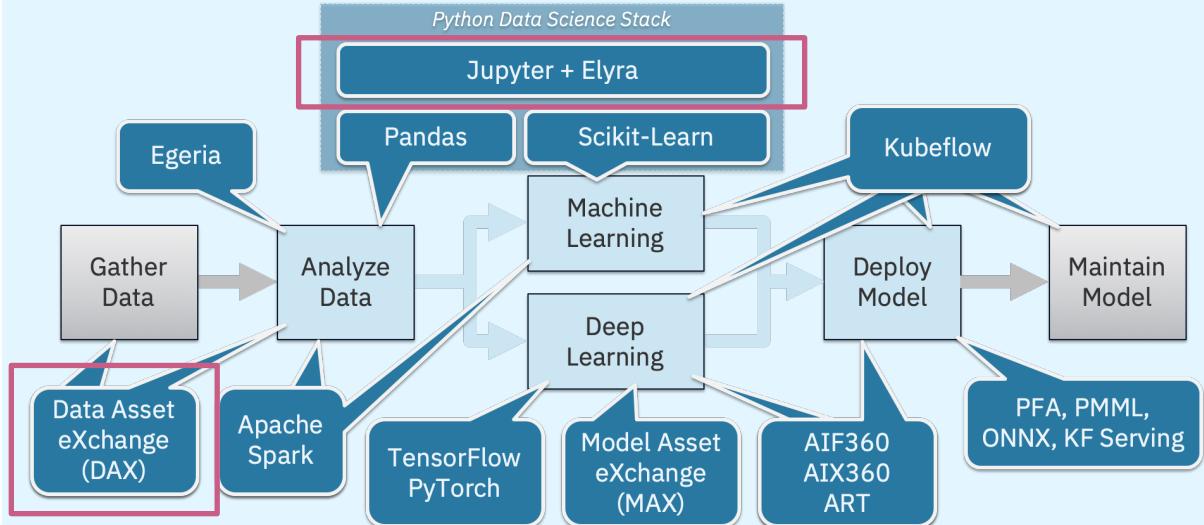
# Overview



- What is CODAIT?
- An overview of the Data Asset eXchange (DAX)
- A deep dive into Elyra and its features
- Demo showcasing DAX & Elyra
- How to get involved

- CODAIT aims to make AI solutions dramatically easier to create, deploy, and manage in the enterprise.
- We contribute to and advocate for the open-source technologies that are foundational to IBM's AI offerings.
- 30+ open-source developers!

#### Improving the Enterprise AI Lifecycle in Open Source



# Data Asset eXchange

Data Asset Exchange offers high-quality datasets with clearly-defined open data licenses in standardized formats, according to IBM.

- Vetted data.
- Exclusive access to IBM Research datasets that have been used in creating popular AI products like [Debater System](#), Entity Recognition, and so on.
- Datasets with open data licenses for both business applications and advancing core science.
- Packaged with tutorials that shows how to read and analyze data. As well as, train machine or deep learning models on IBM Cloud using IBM Cloud AI services as well as multi-cloud AI open-sourced tools.

[ibm.biz/data-exchange](http://ibm.biz/data-exchange)

## Data Asset eXchange

Explore useful and relevant data sets for enterprise data science

[Learn More](#)

What's New



Get Involved



Dataset | CSV

NOAA Weather Data -  
JFK Airport

September 12, 2019

Dataset | IOB format

Groningen Meaning  
Bank - Modified

May 14, 2020

Dataset | CSV

Fashion-MNIST

September 12, 2019

Dataset | JPG, JSON

PubLayNet

October 25, 2019

Dataset | WAV

TensorFlow Speech  
Commands

March 17, 2020

Dataset | PNG, JSON

PubTabNet

November 11, 2019



# Data Preview and Data Glossary

D43 Dataset Preview Notebook Preview Run Notebook in Watson Studio Dataset Homepage

# PubLayNet

Dataset Metadata

Image

Annotated Image (Generated with Notebook)

Dataset Preview

Dataset Glossary

JSON

```
[{"images": [
    {"file_name": "PMC5491543_00004.jpg",
     "height": 794,
     "id": 348952,
     "width": 601},
    {"file_name": "PMC5386292_00002.jpg",
     "height": 792,
     "id": 384435,
     "width": 612},
    {"file_name": "PMC3863508_00003.jpg",
     "height": 792,
     "id": 393156,
     "width": 601}
  ]}, {"annotations": [
    {"segmentation": [
      37, 99,
      360, 34,
      208, 34,
      360, 34,
      288, 46,
      378, 34,
      288, 43,
      378, 34,
      288, 43,
      381, 37,
      288, 43,
      381, 37,
      288, 43,
      391, 26,
      288, 43,
      391, 26
    ]}, {"label": "Background", "bbox_2d": [11, 11, 988, 988], "category": "Background"}, {"label": "Text", "bbox_2d": [11, 11, 988, 988], "category": "Text"}, {"label": "Caption", "bbox_2d": [11, 11, 988, 988], "category": "Caption"}, {"label": "List", "bbox_2d": [11, 11, 988, 988], "category": "List"}, {"label": "Table", "bbox_2d": [11, 11, 988, 988], "category": "Table"}]
```

Critical Care Research and Practice

5

Table 1: Summary of the performance of different models in critical care patients with COVID-19										
	Model	Variable	Performance	Calibration	AUCROC	AMBOB	CORIB	Oncos	Transtute	
1	AMBOB	Age (years)	0.87	0.74	0.81	0.75	0.75	0.82	0.88	0.87
2	AMBOB	Sex (male)	0.86	0.74	0.81	0.75	0.75	0.82	0.88	0.87
3	AMBOB	SpO2 (mmHg)	0.86	0.74	0.81	0.75	0.75	0.82	0.88	0.87
4	AMBOB	ECG (abnormal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
5	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
6	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
7	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
8	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
9	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
10	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
11	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
12	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
13	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
14	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
15	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
16	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
17	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
18	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
19	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
20	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
21	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
22	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
23	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
24	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
25	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
26	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
27	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
28	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
29	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
30	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
31	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
32	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
33	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
34	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
35	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
36	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
37	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
38	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
39	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
40	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
41	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
42	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
43	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
44	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
45	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
46	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
47	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
48	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
49	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
50	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
51	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
52	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
53	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
54	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
55	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
56	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
57	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
58	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
59	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
60	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
61	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
62	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
63	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
64	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
65	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
66	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
67	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
68	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
69	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
70	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
71	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
72	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
73	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
74	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
75	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
76	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
77	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
78	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
79	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
80	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
81	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
82	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
83	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
84	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
85	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
86	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
87	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
88	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
89	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
90	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
91	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
92	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
93	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
94	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
95	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
96	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
97	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
98	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
99	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
100	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
101	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
102	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
103	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
104	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
105	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
106	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
107	AMBOB	ECG (other)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
108	AMBOB	ECG (normal)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
109	AMBOB	ECG (unavailable)	0.85	0.74	0.81	0.75	0.75	0.82	0.88	0.87
110	AMBOB	ECG (other)	0.85	0.74	0					

DAX Dataset Preview		Notebook Preview	Run Notebook in Watson Studio	Dataset Homepage		
PubLayNet						
Dataset Metadata		Dataset Preview		Dataset Glossary		
Feature				Description		
images	JSON field containing a list of images and their metadata (size, ID, name)					
annotations	Each object instance annotation contains a series of fields, including the category id and segmentation mask of the object.					
annotations -> segmentations	Contains the polygon coordinates for the segmentation mask for the specific class instance (table, list, text etc)					
annotations -> bbox	Contains the bounding box coordinates for the specific class instance (table, list, text etc).					
annotations -> is_crowd	This field indicates whether the class instance is a single object ( <code>is_crowd=0</code> ) or multiple objects ( <code>is_crowd=1</code> ). In this dataset we only have single objects so this field is always set to 0.					
annotations -> category_id	The class label for the current class instance. This indicates what the current bbox/segmentation mask encapsulates (table, list, text etc).					
categories	JSON field containing a list of classes and their metadata (ID, name) This dataset has 5 categories (w/ corresponding "ids") - text ("1"), title ("2"), list ("3"), table ("4"), figure ("5").					

[http://bit.ly/pycon\\_elvra](http://bit.ly/pycon_elvra)

# Access notebook in Watson Studio

IBM Cloud Pak for Data

Log In

Sign Up

[Gallery](#) / DAX Weather Project /



[← Back](#)

## DAX Weather Project

Tags

Environment Transportation

Required Services

0

Modified

May 22, 2020

This project includes the NOAA Weather Dataset - JFK Airport (New York) from the Data Asset Exchange and supporting notebooks. The notebooks teach the user to extract, clean and analyze sample weather data and predict weather trends to help airports schedule better flight times. This sample project contains 3 notebooks and 1 CSV file. Please run the notebooks in sequential order of their part numbers using a Python 3.6 runtime.

Images

Assets

Info

# Access from Cloud Pak for Data

The screenshot shows the IBM Cloud Pak for Data product hub interface. At the top, there's a navigation bar with the IBM logo, a search bar, and links for 'What's new', 'Community', and 'Get support'. Below the header, a 'Table of contents' sidebar is visible on the left, listing various sections like 'Overview', 'Use cases', 'Planning', 'Installing', 'Services and integrations', and 'External data sets'. A dropdown menu in the sidebar indicates 'Version 3.0.1 (latest)'. The main content area is titled 'External data sets' and discusses the integration of external data sets for a 360-degree view of business landscapes. It highlights the partnership with industry leaders like The Weather Company for weather data. A table provides details on the 'Weather Company Data Limited Edition' offering, including its provider (The Weather Company), pricing (Included with Cloud Pak for Data), and a 'Learn more' section. The 'Learn more' section includes sections for 'About this offering', 'Use cases', 'Industry accelerators', and a 'Get started' section.

Data offering	Provided by	Pricing	Learn more
Weather Company Data Limited Edition	The Weather Company®	Included with Cloud Pak for Data	<p><b>About this offering</b></p> <p>90-day access to cloud-based APIs that enable you to obtain historical weather data, current conditions, and forecast conditions.</p> <p><b>Use cases</b></p> <p>You can use weather data to optimize operations, reduce overhead costs, increase safety, and uncover new revenue opportunities. For example, you can:</p> <ul style="list-style-type: none"><li>Predict power outages with greater accuracy so that you can restore power to customers faster</li><li>Reduce utility costs with smarter vegetation management</li><li>Improve flight safety, efficiency and performance</li><li>Keep policyholders safe while reducing insurance claims and fraud</li><li>Improve supply chain visibility and minimize weather-related disruptions</li><li>Transport people and goods more safely</li></ul> <p><b>Industry accelerators</b></p> <p>The following industry accelerators can help you get started with this data set:</p> <ul style="list-style-type: none"><li>Manufacturing Analytics with Weather</li><li>Retail Predictive Analytics with Weather</li><li>Sales Prediction using The Weather Company Data</li></ul> <p><b>Get started</b></p> <p>For details, see <a href="https://www.ibm.com/weather">https://www.ibm.com/weather</a>.</p>

[https://www.ibm.com/support/producthub/icpdata/docs/content/SSQNUZ\\_current/svc-nav/data-sets.html](https://www.ibm.com/support/producthub/icpdata/docs/content/SSQNUZ_current/svc-nav/data-sets.html)

[http://bit.ly/pycon\\_elyra](http://bit.ly/pycon_elyra)

# Industrial Accelerator - Cloud Pak for Data

## Cloud Pak for Data

View Only

Group Home    Blogs 0    Members 3

### Effective Farming - Monitor Crop Growth

28 days ago

The accelerator is created using Data Asset eXchange data to support effective farming by monitoring crop growth using crop guide and provide timely alert to farmers about weather change, possible development of crop disease, evaporation of fungicide, and efficient use of solar panels (agrivoltaics support).

#### What's included?

- A structured business glossary of 90 business terms.
- Sample data science assets

#### How does it work?

The glossary provides the information architecture that you need to understand weather related business measures. Your data scientists can use the sample notebooks, predictive models and dashboards to accelerate data preparation, machine learning modeling, and data reporting. Moreover, the data scientists may modify the sample notebooks for other business use cases and corresponding datasets.

Timely alert to farmers can save crop life and bring in more cost savings.

When you import the accelerator:

- The terms are added to your business glossary under the Effective Farming - Monitor Crop Growth category in the Industry Accelerators category.
- The data science assets are added to a new analytics project.

#### Statistics

0 Favorited  
17 Views  
0 Files  
0 Shares  
0 Downloads

<https://community.ibm.com/community/user/cloudpakfordata/viewdocument/effective-farming-monitor-crop-gr>  
[http://bit.ly/pycon\\_elyra](http://bit.ly/pycon_elyra)

# What is Elyra?

Elyra is a set of AI centric extensions to JupyterLab. It aims to help data scientists, machine learning engineers and AI developer's through the model development life cycle complexities.

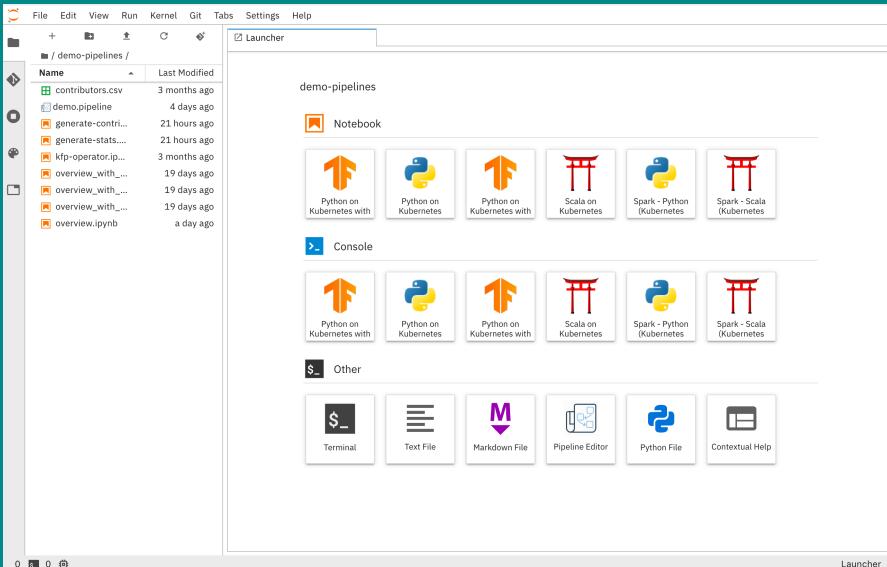


## Elyra on GitHub

<https://github.com/elyra-ai/elyra>

## Elyra's Documentation

<https://elyra.readthedocs.io/en/latest/>

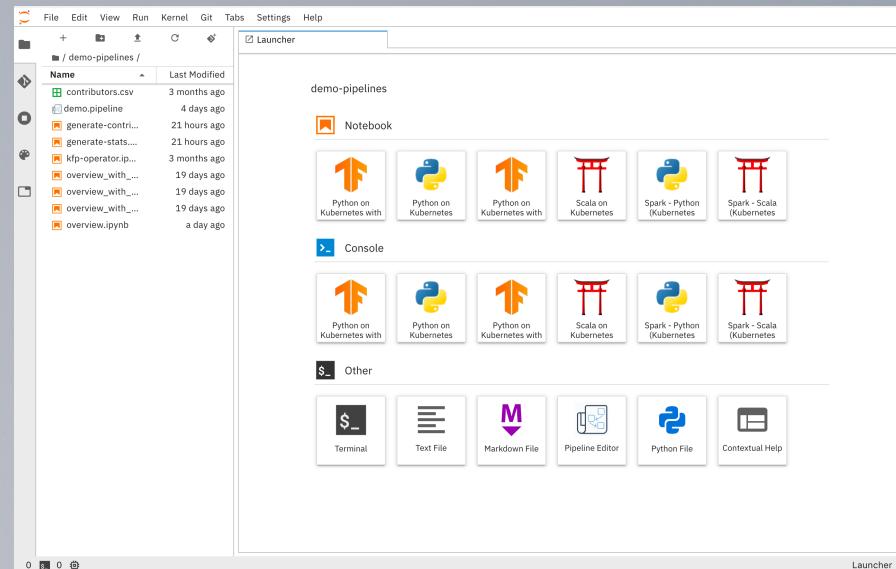


# What is Elyra?

Elyra is a set of  
AI centric extensions for JupyterLab

Elyra was officially announced as an open source project by IBM on April 29<sup>th</sup>.

The name Elyra is a word play with one of the Jupyter moons “Elara” where we introduce the “y” from “Jupyter” to make it “Elyra”



# Elyra Core Features



## Notebook Pipelines editor

Visual editor for building notebook-based AI pipelines, enabling the conversion of multiple notebooks into batch jobs or workflows.

## Notebook as batch jobs

Elyra extends the notebook UI to simplify the submission of notebooks as a batch job for model training

## Code Snippets

Easy creation and insertion of reusable code snippets for the various languages

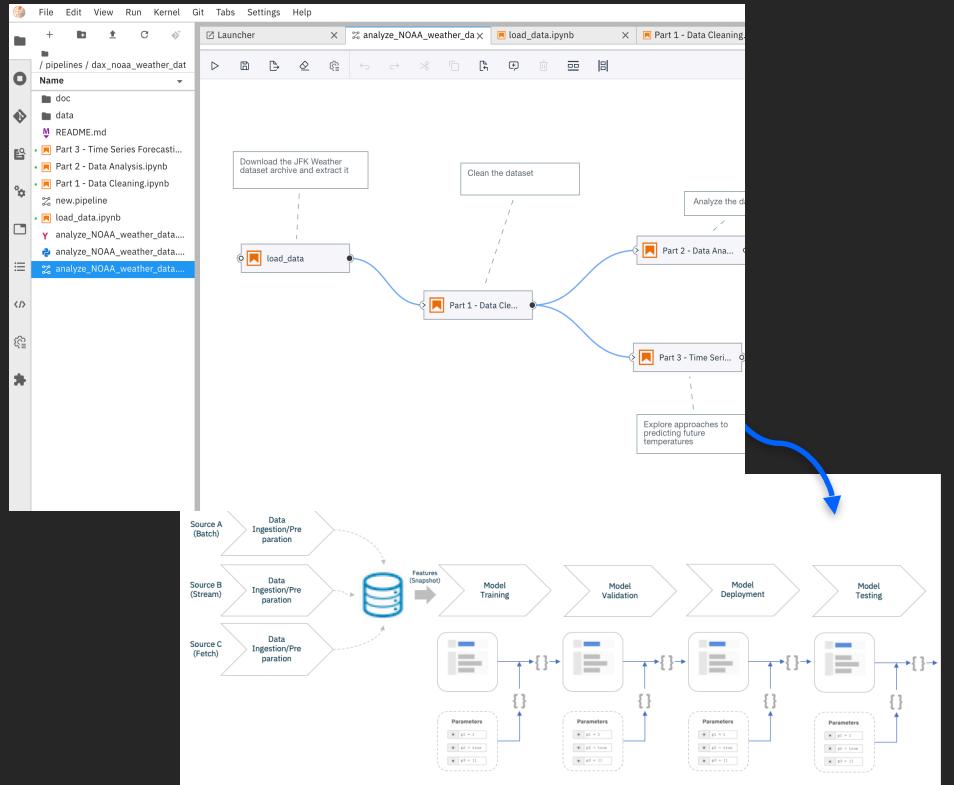
## Git integration

Track project changes and share among teammates

## Python script execution

Edit and execute python scripts against local or cloud-based resources

## Notebook Pipelines



# Elyra Core Features



## Notebook Pipelines editor

Visual editor for building notebook-based AI pipelines, enabling the conversion of multiple notebooks into batch jobs or workflows.

## Notebook as batch jobs

Elyra extends the notebook UI to simplify the submission of notebooks as a batch job for model training

## Code Snippets

Easy creation and insertion of reusable code snippets for the various languages

## Git integration

Track project changes and share among teammates

## Python script execution

Edit and execute python scripts against local or cloud-based resources

## Notebook as batch jobs

The screenshot illustrates the Elyra interface for submitting notebooks as batch jobs. The main window shows a file browser with two entries: 'eu\_data.pipeline' and 'us\_data.pipeline'. In the foreground, a modal dialog box is open, titled 'Submit notebook'. Inside the dialog, there are dropdown menus for 'Runtime Config' (set to 'Kubeflow Pipeline (cloning)') and 'Deep Learning Framework' (set to 'Tensorflow 2.0'). There is also a checked checkbox for 'Include dependencies'. Below the dialog, a Python code snippet is displayed:

```
plt.figure(figsize=(8,8))
plt.scatter(x,y)
plt.plot(line_x, line_y, color='red', lw=2)

plt.xlabel('x')
plt.ylabel('y')
plt.title('Linear regression via least squares fit')

ftext = 'y = ax + b = (%.3f) + (%.3f)x\n' % (slope, intercept)
plt.text(100, 100, ftext, transform=plt.gca().transAxes)
```

Below the code, a scatter plot is shown with a red line representing a linear regression fit. The plot has axes labeled 'x' and 'y'.



# Elyra Core Features

## Notebook Pipelines editor

Visual editor for building notebook-based AI pipelines, enabling the conversion of multiple notebooks into batch jobs or workflows.

## Notebook as batch jobs

Elyra extends the notebook UI to simplify the submission of notebooks as a batch job for model training

## Code Snippets

Easy creation and insertion of reusable code snippets for the various languages

## Git integration

Track project changes and share among teammates

## Python script execution

Edit and execute python scripts against local or cloud-based resources

## Code Snippets

The screenshot shows the Elyra Notebook Pipelines editor interface. On the left, there's a sidebar titled 'Code Snippets' containing a tree view of available snippets categorized by language (C/C++, Scala, Python) and scenario (Spark - Bank Scenario, Read Environment Variable, Spark - Configuration details). A snippet for 'python|Matplotlib simple graph' is selected. The main workspace on the right displays a Jupyter Notebook cell with the following Python code:

```
[1]: from __future__ import print_function, division
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline

[2]: # Silly example data
bp_x = np.linspace(0, 2*np.pi, num=40, endpoint=True)
bp_y = np.sin(bp_x)

# Make the plot
plt.plot(bp_x, bp_y, linewidth=2, linestyle='--',
         color='blue', label="Legend label sin(x)")
plt.xlabel("Description of x coordinate (units)")
plt.ylabel("Description of y coordinate (units)")
plt.title("Title here (remove for papers)")
plt.xlim(0, 2*np.pi)
plt.ylim(-1.1, 1.1)
plt.legend(loc='lower left')
plt.show()
```

Below the code, a plot is generated showing a sine wave from 0 to 2π. The x-axis is labeled "Description of x coordinate (units)" and the y-axis is labeled "Description of y coordinate (units)". The plot title is "Title here (remove for papers)". The legend indicates "Legend label sin(x)".

# Elyra Core Features



## Notebook Pipelines editor

Visual editor for building notebook-based AI pipelines, enabling the conversion of multiple notebooks into batch jobs or workflows.

## Notebook as batch jobs

Elyra extends the notebook UI to simplify the submission of notebooks as a batch job for model training

## Code Snippets

Easy creation and insertion of reusable code snippets for the various languages

## Git integration

Track project changes and share among teammates

## Python script execution

Edit and execute python scripts against local or cloud-based resources

## Git integration

The screenshot shows the Elyra Jupyter Enterprise Gateway interface. On the left, there is a file browser titled "Current Repository" showing a "sample-notebooks" folder, a "master" branch, and a "Changes" tab with one staged file named "generate-contributions.ipynb". In the center, there is a terminal window titled "generate-contributions.ipynb" with the following code:

```
In [1]: !pip install PyGithub pandas >/dev/null 2>61
In [2]: import os
In [3]: import datetime
In [4]: import pandas as pd
In [5]: from github import Github
In [6]: github = Github(os.environ['GITHUB_TOKEN'])
```

Below the terminal, there is a section titled "Jupyter Enterprise Gateway Contribution Stats" with some sample code:

```
In [1]: github_jup_org = github.get_organization('jupyter')
github_jup_repo = github_jup_org.get_repo('enterprise_gateway')
In [2]: datetime_start = datetime.datetime.now() + datetime.timedelta(days=-180) #datetime.datetime(2018, 7, 1)
In [3]: contributions_df.to_csv('community_contributions.csv', index=False)
```

At the bottom, there is a "Commit" button and a "Summary (required)" field with "Description" and "Commit" sections.

# Elyra Core Features



## Notebook Pipelines editor

Visual editor for building notebook-based AI pipelines, enabling the conversion of multiple notebooks into batch jobs or workflows.

## Notebook as batch jobs

Elyra extends the notebook UI to simplify the submission of notebooks as a batch job for model training

## Code Snippets

Easy creation and insertion of reusable code snippets for the various languages

## Git integration

Track project changes and share among teammates

## Python script execution

Edit and execute python scripts against local or cloud-based resources

## Python Script editor

The screenshot shows the Elyra Python Script editor interface. On the left is a file browser with a folder named 'PANDA.py' containing code for importing io, requests, pandas, and time, along with a function 'delay(seconds)'. To the right is a code editor window titled 'panda.py' with the same code. Below the code editor is a terminal window titled 'Python Console Output' showing the execution of a script that reads data from a URL and calculates the mean price per zipcode. The output shows several rows of data and a final summary line: 'Name: price, Length: 68, dtype: float64'.

```
1 # Add sample panda code to manipulate the generated df
2 import io
3 import requests
4 import pandas as pd
5 import time
6
7 def delay(seconds):
8     time.sleep(seconds)
9
10 def df_from_url(url):
11     data = requests.get(url).content
12     df = pd.read_csv(io.StringIO(data.decode('utf-8')))
13     return df
14
15 # Uncomment the lines below to sleep for a bit
16 # useful to demonstrate kernel startup on container environments
17 # delay(3)
18
19 # Sample panda code to manipulate the generated data frame
20 # and calculate mean price per zipcode
21 df = df_from_url('http://samplecsvs.s3.amazonaws.com/SacramentoRealEstateTransactions.csv')
22 df.groupby(['zip'])['price'].mean()
```

```
[1]: zip
95603    405500.000000
95608    795084.750000
95610    226436.285714
95614    300000.000000
95619    216033.000000
95838    149461.351351
95841    213806.142857
95842    143281.772727
95843    227000.000000
95864    364400.000000
Name: price, Length: 68, dtype: float64
```

# Data Science Process

Data Extraction

Data Cleaning

Data Exploration

Model Development

Result Interpretation

# Getting Started



## What are the pre-requisites to run?

- NodeJS 12+
- Python 3.X
- Anaconda (optional)
- JupyterLab Support
- KubeFlow Installation (optional)

# Install Elyra



To install Elyra:

```
$ pip install --upgrade elyra && jupyter lab build
```

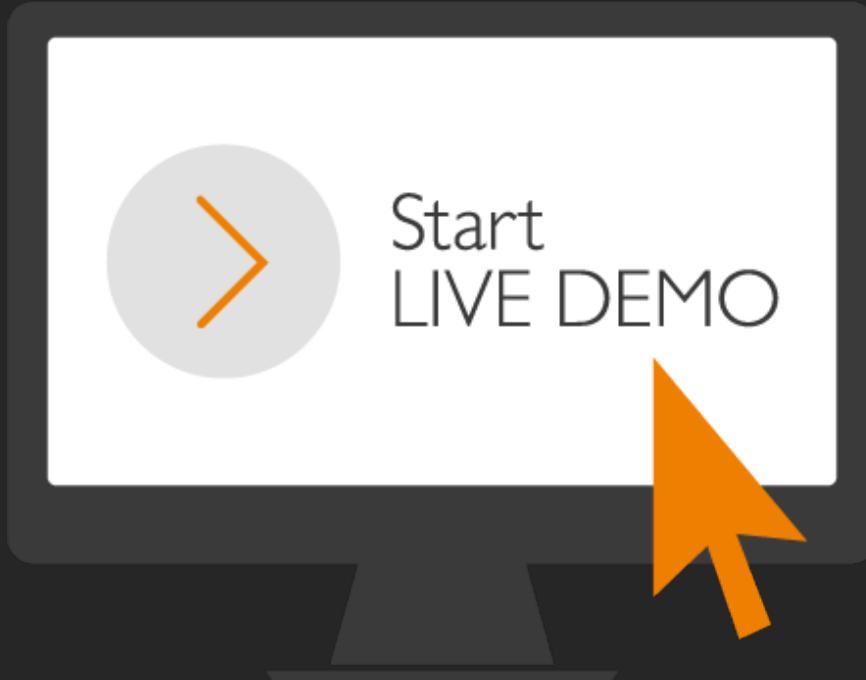
To verify installation:

```
$ jupyter serverextension list And
```

```
$ jupyter labextension list
```

Starting Elyra:

```
$ jupyter lab
```



# Get involved

## Getting started with Elyra

[https://elyra.readthedocs.io/en/latest/getting\\_started/installation.html](https://elyra.readthedocs.io/en/latest/getting_started/installation.html)



## Elyra's Github

<https://github.com/elyra-ai/elyra>

## Data Asset eXchange

<https://developer.ibm.com/exchanges/data/>

## DAX notebooks Github

[https://github.com/elyra-ai/examples/tree/master/pipelines/dax\\_noaa\\_weather\\_data](https://github.com/elyra-ai/examples/tree/master/pipelines/dax_noaa_weather_data)

## Contributing to these projects

- Bug reports
- Enhancement requests
- Code reviews

# Data Asset eXchange

Explore useful and relevant data sets for enterprise data science

