

How to perform Data Science Analysis using Elyra?



Yiwen Li

About us



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



Yiwen Li
Data Scientist &
Developer Advocate



Yiwen.Li@ibm.com



github.com/yil532



linkedin.com/in/yiwenli

Overview



- Data Science Pipeline
- 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

Data Science Process

Data Extraction

Data Cleaning

Data Exploration

Model Development

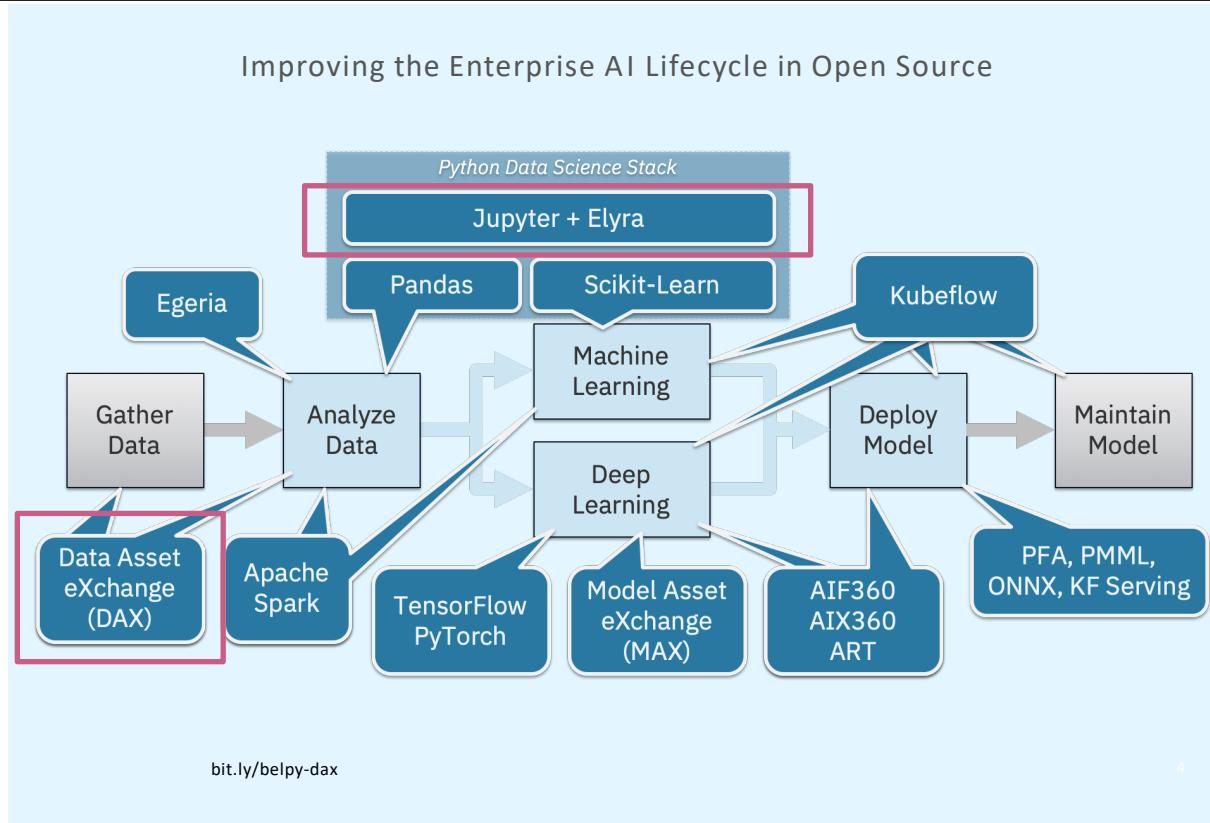
Result Interpretation

CODAIT

Open Source @IBM



- 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!



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

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

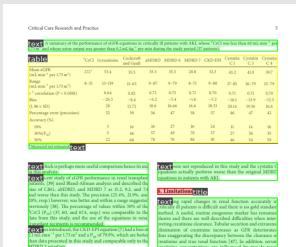
DAX Dataset Preview | Notebook Preview | Run Notebook in Watson Studio | Dataset Homepage

PubLayNet

Dataset Metadata

Dataset Preview

Dataset Glossary



```
JSON
[{"images": [
  {"file_name": "PMC5491943_00004.jpg",
   "height": 794,
   "id": 349952,
   "width": 570},
  {"file_name": "PMC5382692_00002.jpg",
   "height": 792,
   "id": 384435,
   "width": 612},
  {"file_name": "PMC3863500_00003.jpg",
   "height": 792,
   "id": 384436,
   "width": 603}
],
"annotations": [
  {"segmentation": [
    [37.99,
     388.24,
     288.66,
     360.34,
     298.45,
     370.34,
     288.43,
     370.24,
     288.43,
     381.37,
     280.24,
     381.37,
     288.43,
     371.24,
     272.66]
  ]
}]}]
```

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 (is_crowd=0) or multiple objects (is_crowd=1). 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").

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'. The main content area displays the 'External data sets' page, which includes a breadcrumb trail ('IBM Cloud Pak for Data > Services and integrations'), a section about external data sets, and a table comparing different offerings. The table has columns for 'Data offering', 'Provided by', 'Pricing', and 'Learn more'. One row is shown for 'Weather Company Data Limited Edition' provided by 'The Weather Company®' with 'Included with Cloud Pak for Data'. The 'Learn more' section for this offering includes details about historical weather data, current conditions, and forecast conditions, along with a 'Use cases' section listing various applications such as optimizing operations, reducing overhead costs, and improving safety. There are also sections for 'Industry accelerators' and 'Get started'.

Data offering	Provided by	Pricing	Learn more
Weather Company Data Limited Edition	The Weather Company®	Included with Cloud Pak for Data	<p>About this offering</p> <p>90-day access to cloud-based APIs that enable you to obtain historical weather data, current conditions, and forecast conditions.</p> <p>Use cases</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">Predict power outages with greater accuracy so that you can restore power to customers fasterReduce utility costs with smarter vegetation managementImprove flight safety, efficiency and performanceKeep policyholders safe while reducing insurance claims and fraudImprove supply chain visibility and minimize weather-related disruptionsTransport people and goods more safely <p>Industry accelerators</p> <p>The following industry accelerators can help you get started with this data set:</p> <ul style="list-style-type: none">Manufacturing Analytics with WeatherRetail Predictive Analytics with WeatherSales Prediction using The Weather Company Data <p>Get started</p> <p>For details, see https://www.ibm.com/weather.</p>

https://www.ibm.com/support/producthub/icpdata/docs/content/SSQNUZ_current/svc-nav/data-sets.html

bit.ly/belpy-dax

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>

bit.ly/belpy-dax

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.

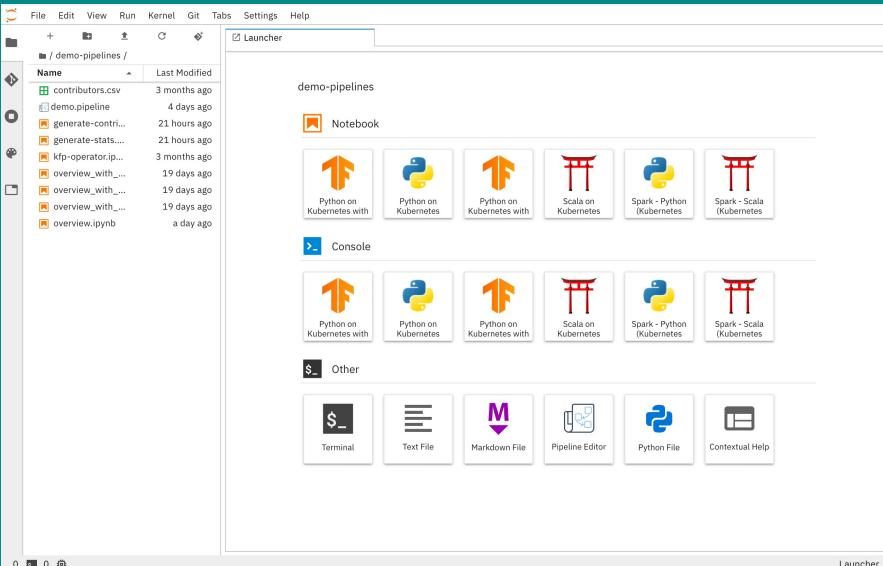
Elyra on GitHub

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



Elyra's Documentation

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

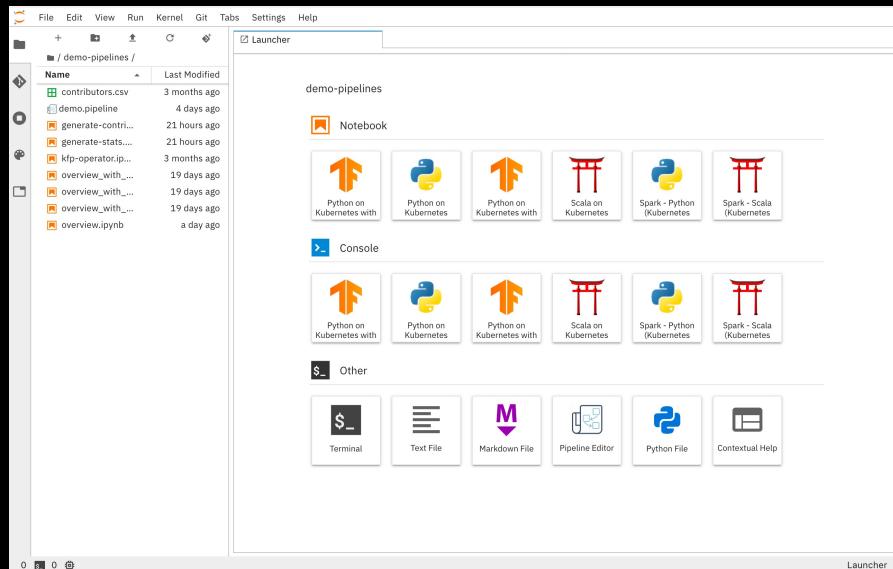


What is Elyra?

Elyra is a set of
AI centric extensions for JupyterLab

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

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

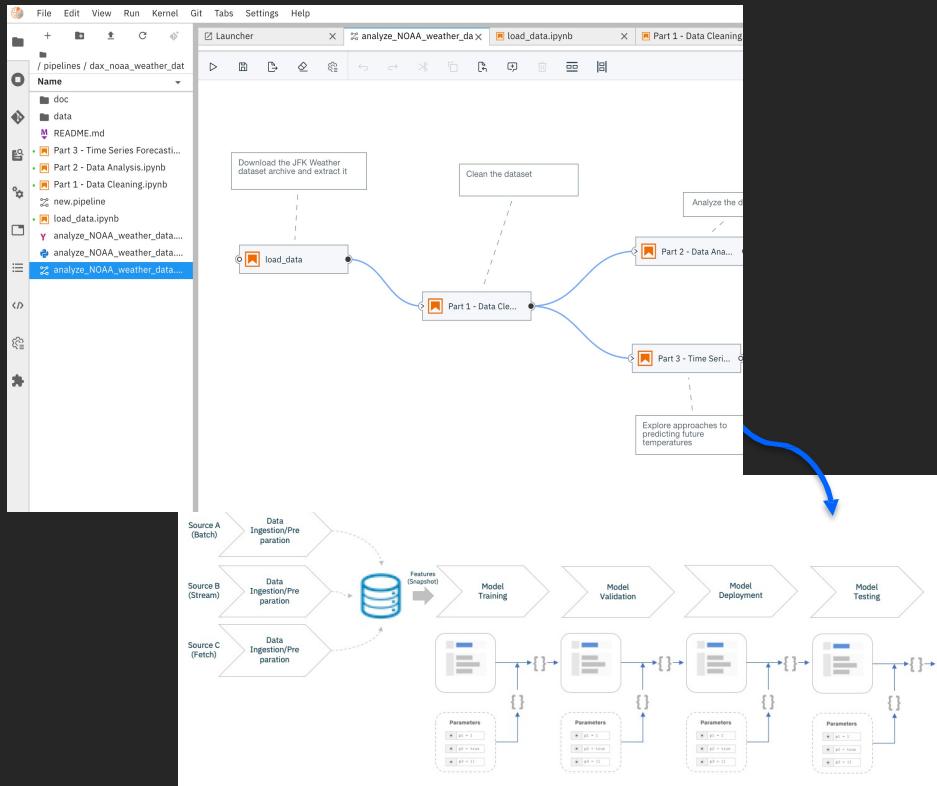
Git integration

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 'Notebook as batch jobs' feature. In the background, a Jupyter Notebook interface is shown with multiple tabs: 'demo-new.pipeline', 'demo.pipeline', 'us_data.pipeline', 'demo-new.py', and 'python-linregr-least-squares.ipynb'. The 'python-linregr-least-squares.ipynb' tab is active, displaying Python code for performing a linear regression fit on a dataset. A modal dialog box is overlaid on the interface, titled 'Submit notebook'. It contains a dropdown menu for 'Runtime Config' set to 'Kubeflow Pipeline (cloning)', a dropdown for 'Deep Learning Framework' set to 'Tensorflow 2.0', and a checked checkbox for 'Include dependencies'. At the bottom of the dialog are 'Cancel' and 'OK' buttons. The main content area shows a scatter plot with a red regression line.



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 Code Snippets interface. On the left, there's a sidebar with a tree view of code snippets categorized by language: C/CPP, Scala, Python, and Java. The Python section is expanded, showing snippets like 'My cpp code', 'Spark - Environment Variable', 'Read Environment Variable', 'Spark - Configuration details', and 'Mapplotlib Configuration'. On the right, a code editor window titled 'Untitled.ipynb' displays Python code for generating a sine wave plot. The code includes imports for numpy, matplotlib, and matplotlib.pyplot, and uses np.linspace to create x and y arrays. A plot window below the code shows a blue dashed sine wave from x=0 to x=6. The x-axis is labeled 'Description of x coordinate (units)' and the y-axis is labeled 'Description of y coordinate (units)'. The plot has a title 'Title here (remove for papers)' and a legend entry 'Legend label sin(x)'.

```
[1]: # free futures
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=3, 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(-1.1, 1.1)
plt.ylim(-1.1, 1.1)
plt.legend(loc='lower left')
plt.show()
```

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 Jupyter Enterprise Gateway Contribution Stats interface. At the top, there's a 'Launcher' tab and a file named 'generate-contributions.ipynb'. Below it, a 'Changes' section shows a single file 'generate-contributions.ipynb' has been modified. The main area contains a code editor with Python code for interacting with GitHub:

```
In [1]: pip install PyGithub pandas >/dev/null 2>&1
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'])

In [7]: contributions_df.to_csv('community_contributions.csv', index=False)
```

Below the code editor, there's a 'Commit' dialog with fields for 'Summary (required)' and 'Description', and a 'Commit' button at the bottom.

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 Scripteditor

The screenshot shows the Python Scripteditor interface. On the left is a file browser with files PANDA.py and io.py. The main area is a code editor with the following Python script:

```
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()
```

Below the code editor is a Python Console Output window showing the results of the script execution:

```
[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
```

Getting Started



What are the pre-requisites to run?

- NodeJS 12+
- Python 3.X
- Anaconda (optional)
- JupyterLab Support
 - JupyterLab 1.X is supported on Elyra 0.10.x and below
 - JupyterLab 2.X is supported on Elyra 1.0.0 and above
- KubeFlow Installation (optional)

Install Elyra



To install Elyra:

```
$ pip install elyra==1.1.0 && jupyter lab build Or:
```

```
$ pip install --upgrade elyra && jupyter labbuild
```

To verify installation:

```
$ jupyter serverextension list And
```

```
$ jupyter labextension list
```

Starting Elyra:

```
$ jupyter lab
```



Start
LIVE DEMO



Get involved



Getting started with Elyra

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

Contributing to these projects

- Bug reports
- Enhancement requests
- Code reviews

Data Asset eXchange

Explore useful and relevant data sets for enterprise data science

Related Links



Slides: http://bit.ly/pycon_elyra

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

DAX Asset eXchange: <http://ibm.biz/data-exchange>

Elyra demo Github: <https://github.com/elyra-ai/examples/>

Sign up for IBM Cloud: <https://ibm.biz/BdqVxW>

Model Asset eXchange: <http://ibm.biz/model-exchange>

