Getting started with the Model Asset Exchange and the Data Asset Exchange

In this lab you will explore the Model Asset Exchange (MAX) and the Data Asset Exchange (DAX), which are two open source Data Science resources on IBM Developer.

Objective for Exercise 1:

- Find ready-to-use deep learning models on the Model Asset Exchange
- Locate resources that guide you through deployment in a local or cloud environment
- Explore the deep learning model-serving microservice API using your web browser
- Articulate how developers can consume those microservices

Objective for Exercise 2

- Where to find open data sets on IBM Developer
- How to explore those data sets

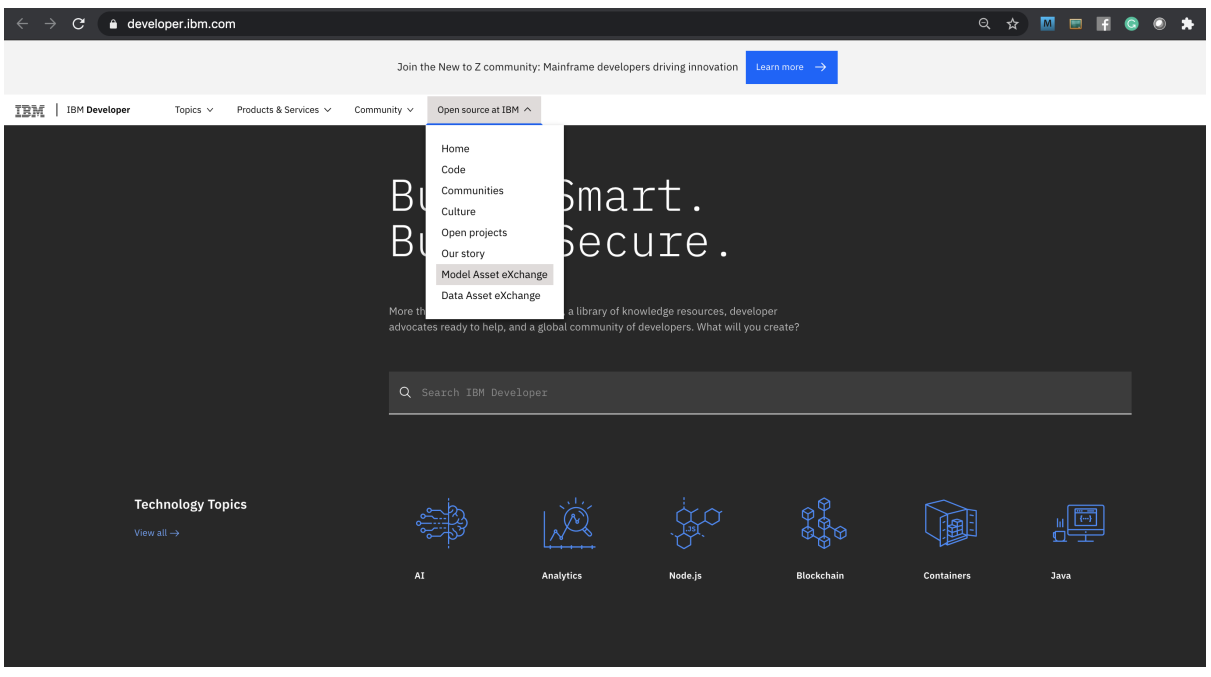
It will take you approximately 30 minutes to complete the lab. Only a web browser is required to complete the tasks.

Exercise 1 - Explore deep learning models

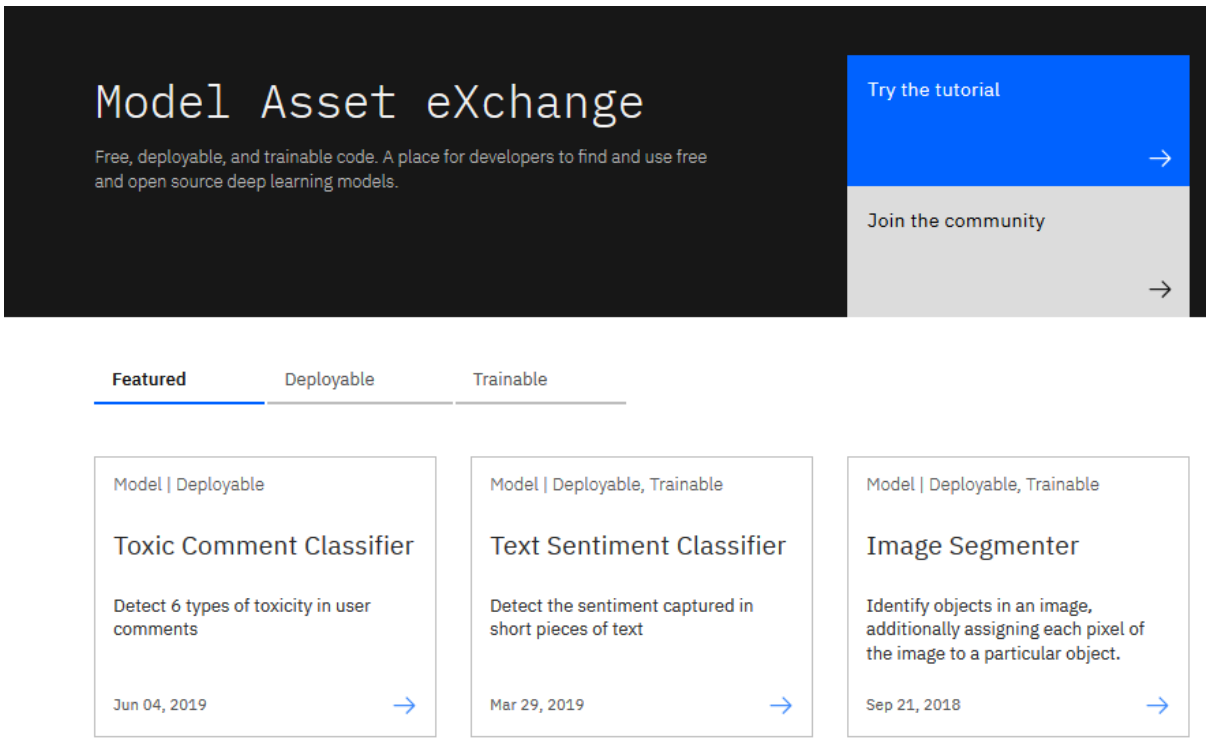
The Model Asset Exchange is a curated repository of open source deep learning models for a variety of domains, such as text, image, audio, and video processing.

1. Open https://developer.ibm.com/ in your web browser.

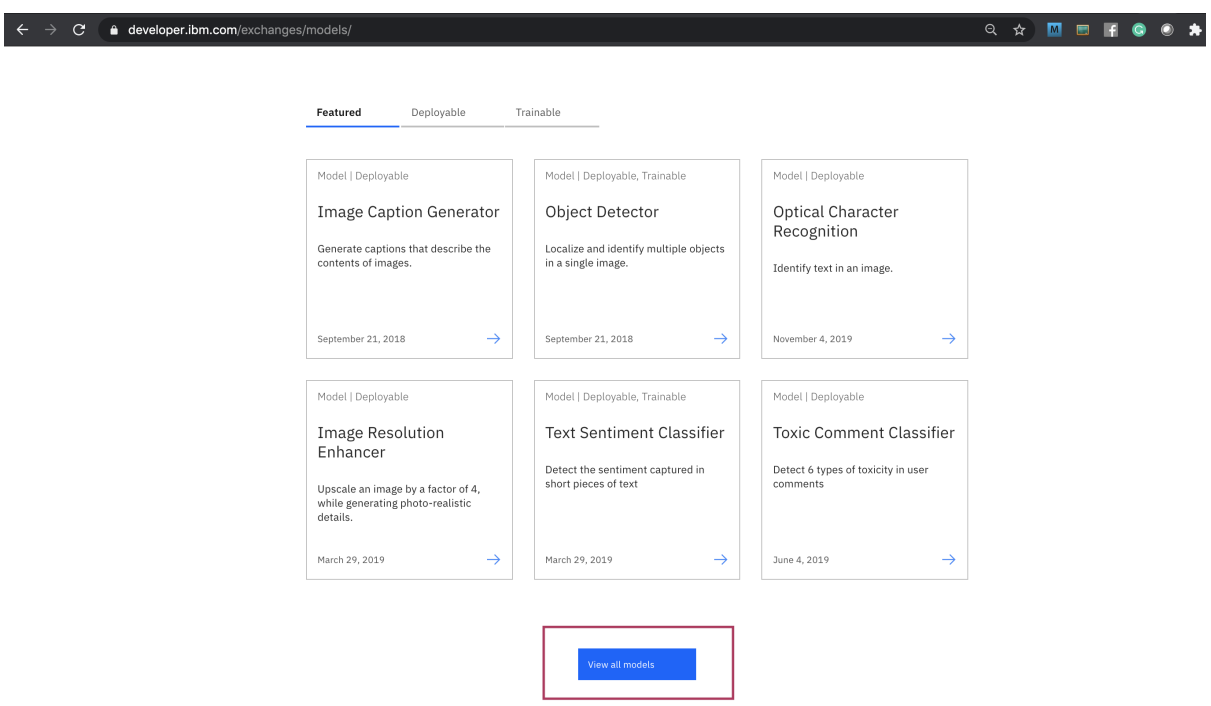
2. From the main menu select **“Open Source at IBM” > “Model Asset eXchange”.**



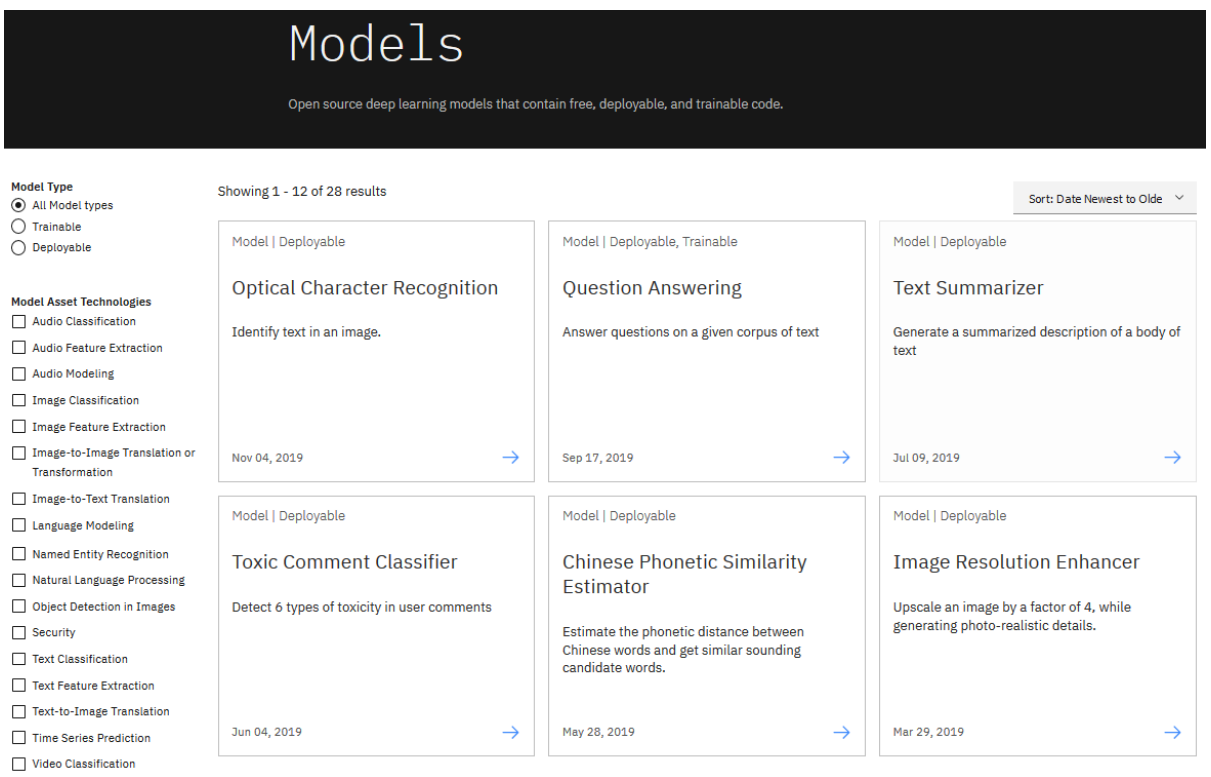
3. The MAX home page is displayed. *In this introductory lab exercise, we are going to focus on a few MAX key features. More detailed information is available in the Learning Path, which covers common deployment and consumption scenarios.*



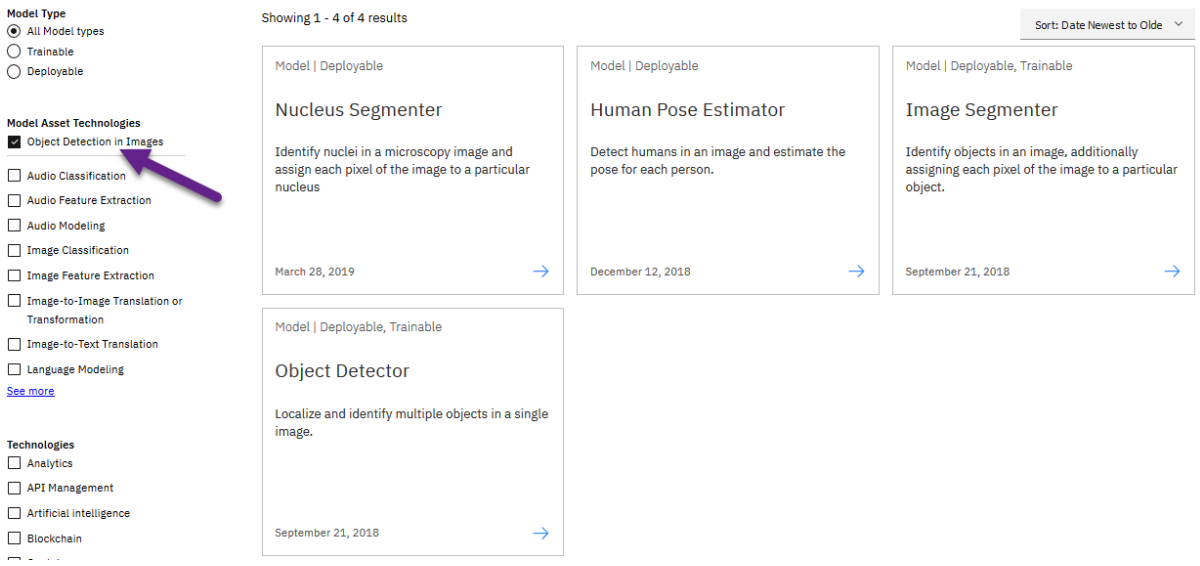
4. Scroll down the page and click on **View all Models**.



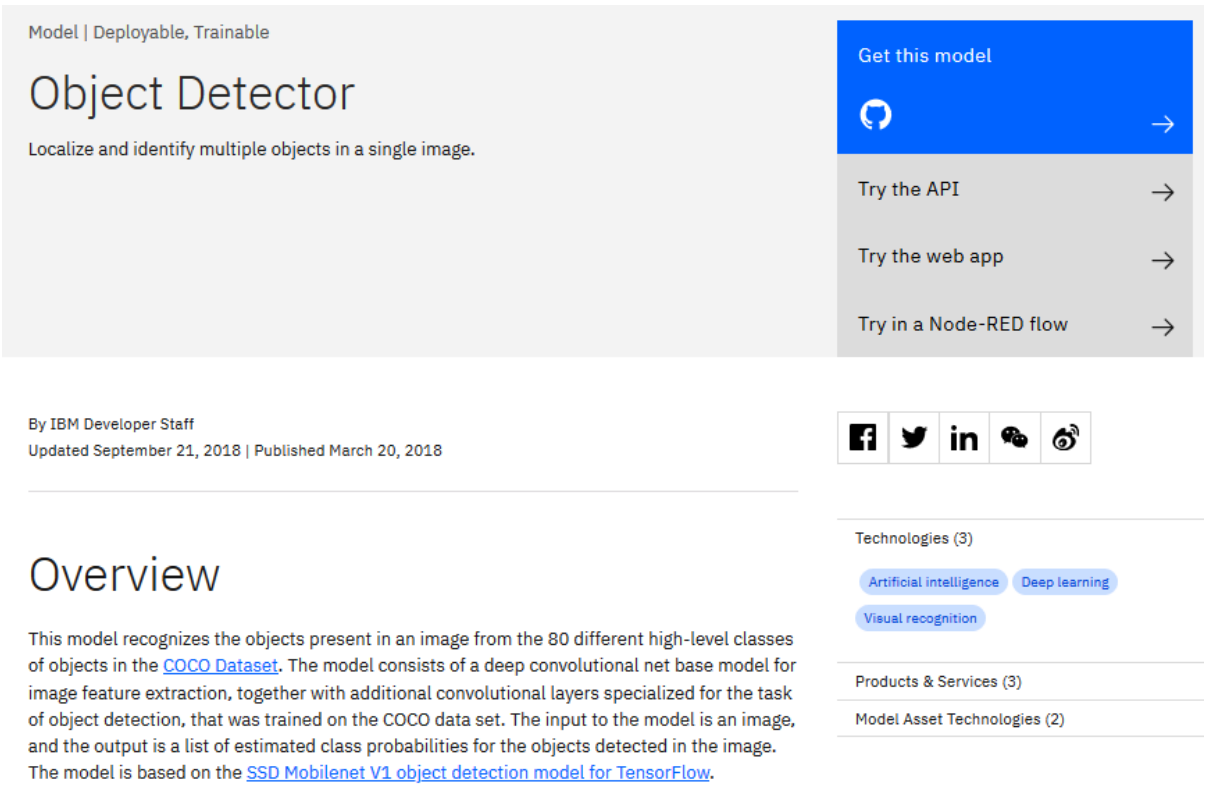
A page with all the models is displayed.



5. Select **Object Detection in Images** from the *Model Asset Technologies* list on the left side.

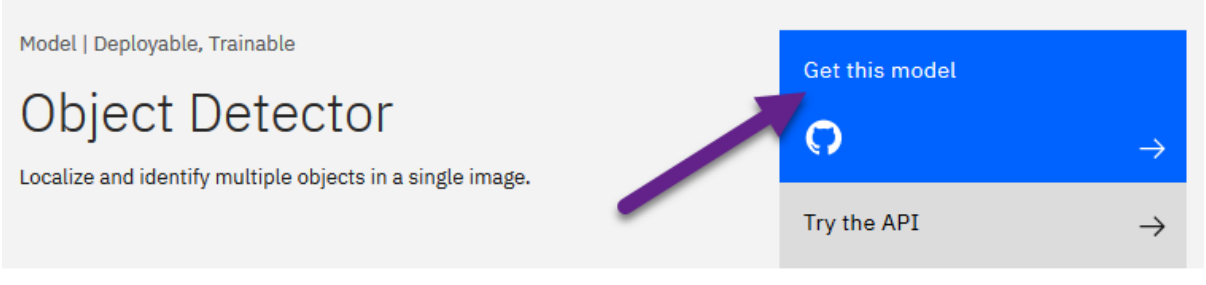


6. Four models should be displayed: Nucleus Segmenter, Human Pose Estimator, Image Segmenter, and Object Detector. Select the **Object Detector model.**



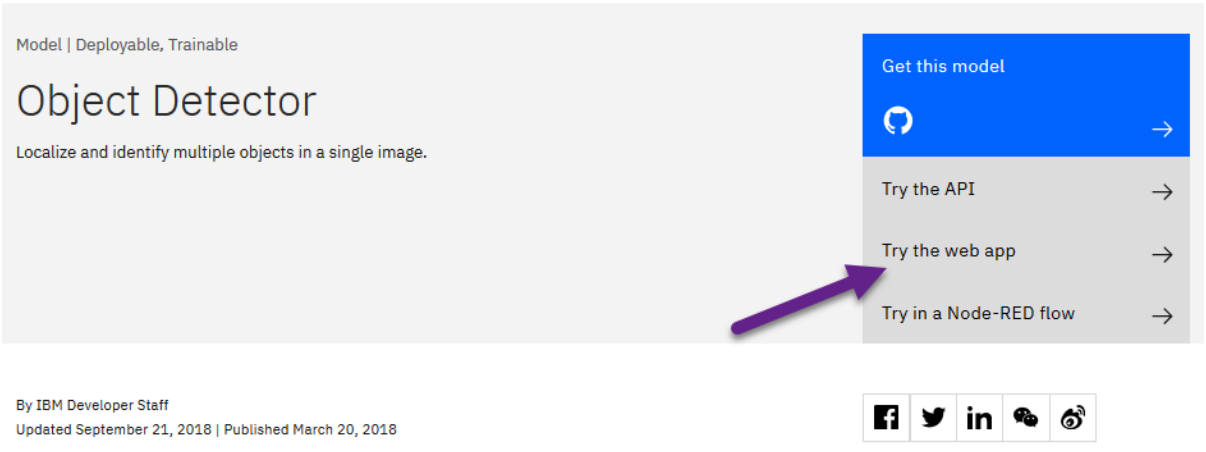
On the model page you can learn about the underlying technology, read up on related research, and explore deployment and consumption options.

7. The model's source is published on GitHub and can be downloaded and modified if desired. Click "Get this model" to open the repository in a new browser tab (or window) and view the files. You can use this once you are comfortable deploying models.

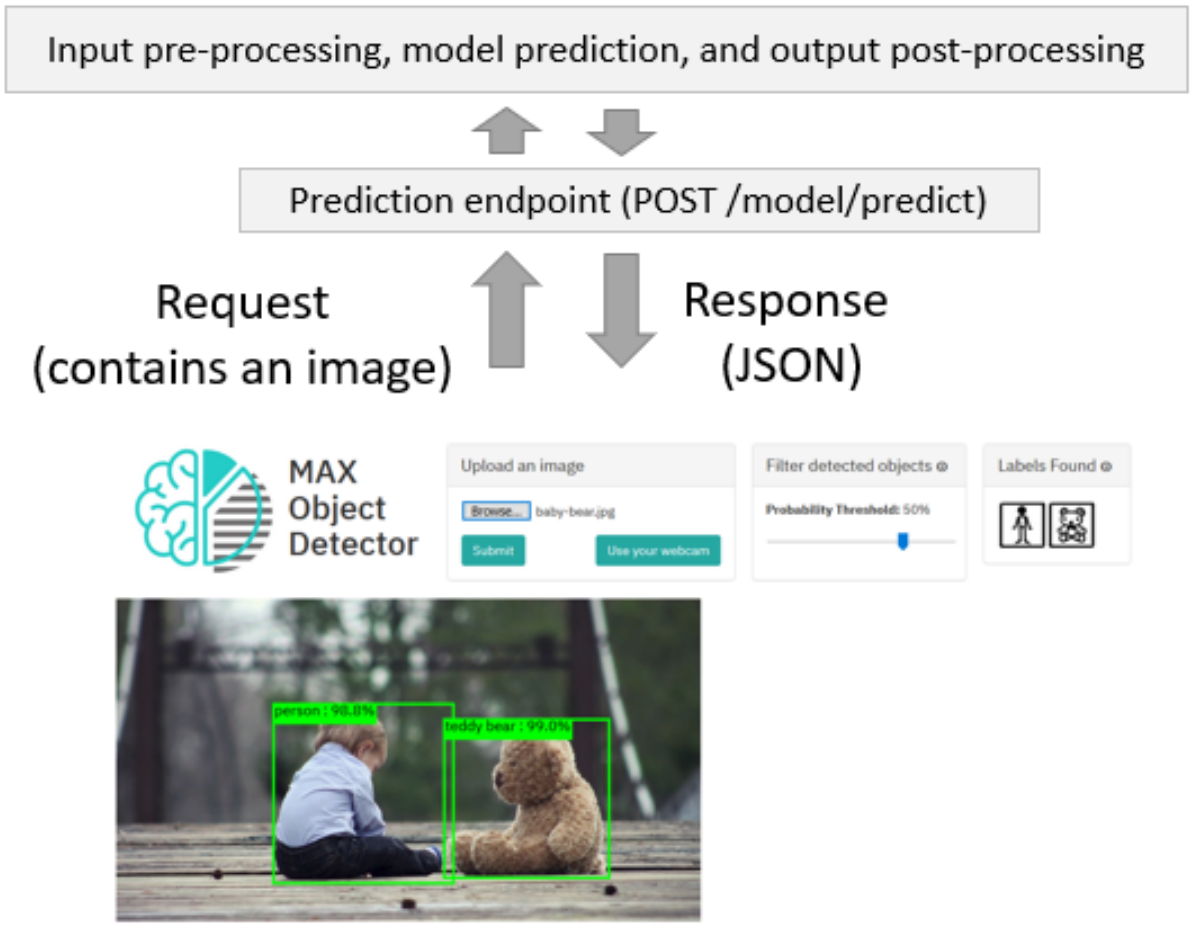


8. Close the GitHub window or tab

9. On the Object Detector page (https://developer.ibm.com/exchanges/models/all/max-object-detector/) on top right, choose **Try the web app**.



The demo web application uploads an image (or takes a picture using the web cam), sends a request to an Object Detector microservice and visualizes the response by drawing bounding boxes around detected objects and attaching a label.



Upload an image (or take a picture using the web camera) and inspect the visualized results.

10. Change the filter conditions (e.g. lower the probability threshold) and observe how they impact the visualized results. Note the web application caches the response and applies the filter on the cached data. This was done for two reasons:

- It eliminates the need to re-process the image using the deep learning microservice (which significantly lowers the application's response time)
- It illustrates how client-side filtering can be applied to the results if the prediction endpoint doesn't support filtering by the desired criteria.

If you are interested in learning more about how the application was implemented, take a look at the code pattern: https://developer.ibm.com/patterns/create-a-web-app-to-interact-with-objects-detected-using-machine-learning/

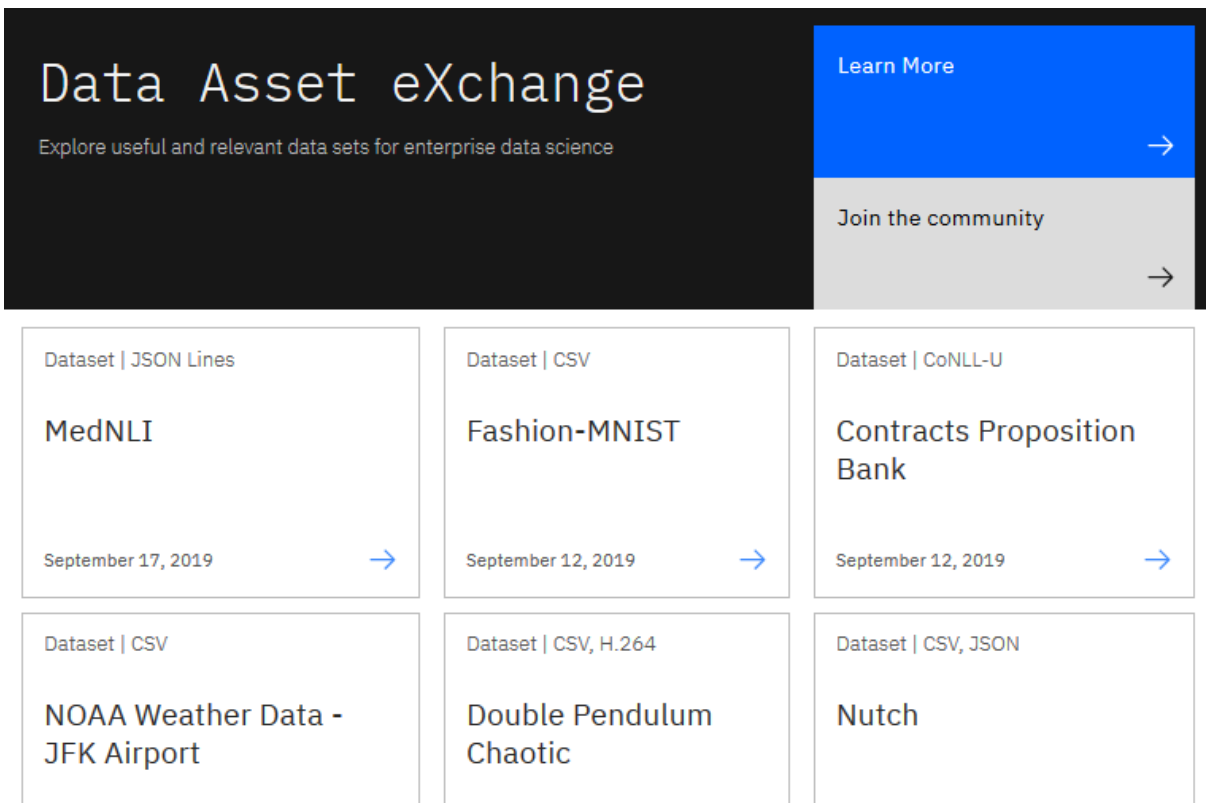
This concludes Exercise 1 of this lab, which introduced the Model Asset Exchange.

Exercise 2: Explore deep learning datasets

The Data Asset Exchange is a curated collection of open datasets from IBM Research and 3rd parties that you can use to train models.

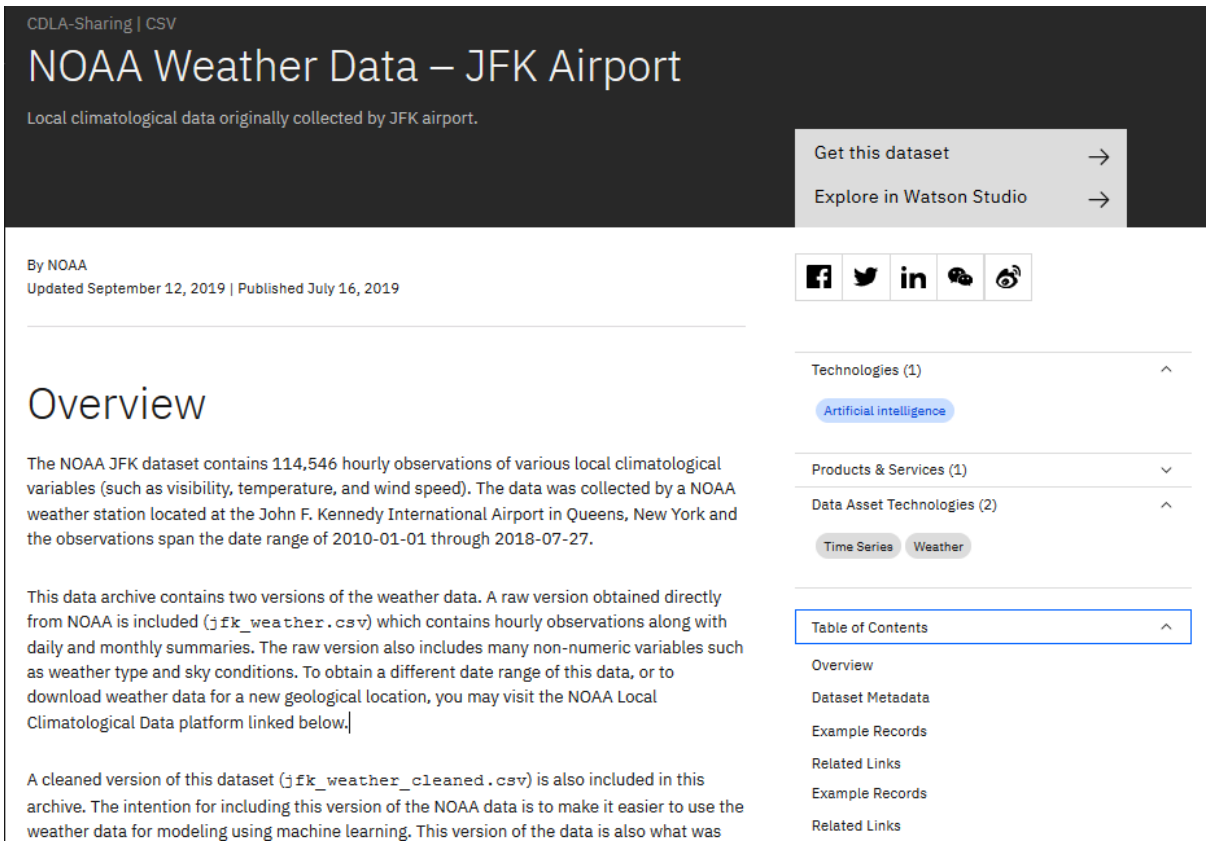
1. Open https://developer.ibm.com/ in your web browser.

2. From the main menu select "Open Source at IBM" > "Data Asset eXchange". The DAX homepage is displayed.



The collection includes datasets from the Debater project (https://www.research.ibm.com/artificial-intelligence/project-debater/), datasets that can be used to train models to perform document layout analysis, natural language processing, time series analysis and more.

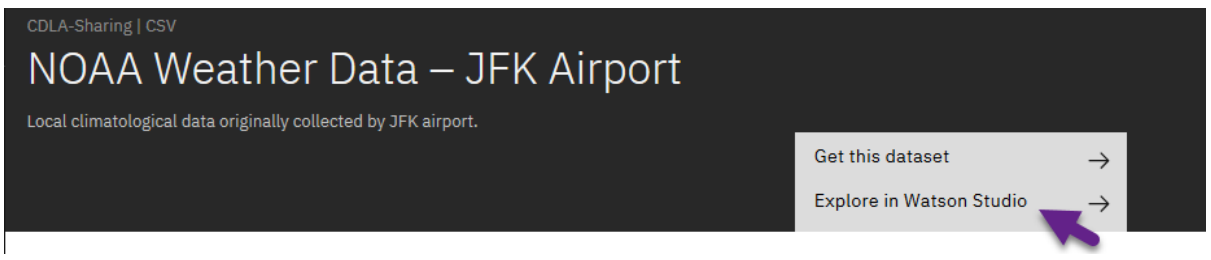
3. Open the NOAA Weather Data dataset (https://developer.ibm.com/exchanges/data/all/jfk-weather-data/), which contains data from a weather station at the John F. Kennedy Airport in New York spanning 8 years. This dataset was used to train the weather forecaster model on MAX (https://developer.ibm.com/exchanges/models/all/max-weather-forecaster/).



You can download the dataset using the “Get this dataset” link. Datasets are stored as compressed archives, which you can extract using any utility that supports the tar.gz format. If you are not familiar with this file format take a look at this short open source tutorial https://opensource.com/article/17/7/how-unzip-targz-file.

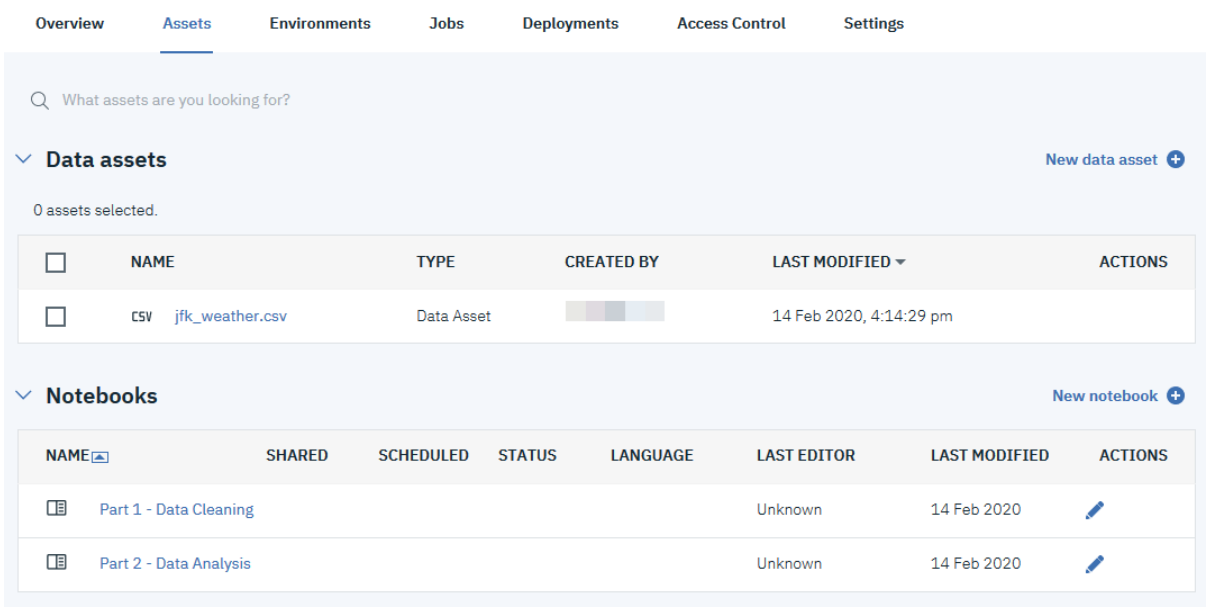
4. Inspect the dataset’s metadata.

This dataset is stored as tabular data and formatted as a comma separated value (CSV) file, which is a very popular basic data exchange format. The dataset was published under the data science friendly CDLA-Sharing license (https://cdla.io/). The dataset contains time-series data and can be used to predict weather trends. 5. Most datasets are complemented by Python notebooks that you can use to explore, pre-process, and analyze the data. You can access the notebook (or notebooks) by clicking the “Explore in Watson Studio” or “Try the notebook” link:



The notebooks are hosted on Watson Studio, IBM’s Data Science platform. Later in this course you’ll learn more about Watson Studio, notebooks and how to run them.

6. [Optional] If you are already familiar with notebooks and Watson Studio feel free to open the link and import the project or notebook. The example below depicts the weather dataset project assets, which include the raw data file and two notebooks.



This concludes Exercise 2 of this lab, which introduced the Data Asset Exchange.

Author(s)

Joseph Santarcangelo

Other Contributor(s)

Lavanya

Changelog

Date	Version	Changed by	Change Description
2020-08-25	2.0	Lavanya	Migrated Lab to Markdown and added to course repo in GitLab