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 of Exercise 1:

- Find open data sets on IBM Developer.
- Explore the data sets.

Objective of Exercise 2:

- Find ready-to-use deep learning models on the Model Asset Exchange.
- Explore the deep learning model trained to detect objects in an image.

It will take you approximately 15 minutes to complete the lab. Only a web browser is required to complete the tasks.

Exercise 1: Explore deep learning datasets

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

1. Open https://developer.ibm.com/exchanges/data/ in your web browser.The IBM Data Asset eXchange (DAX) home page is displayed. This is an online hub for developers and data scientists to find free and open data sets under open data licenses. These datasets can be used to train models to perform document layout analysis, natural language processing (NLP), time series analysis, and more.
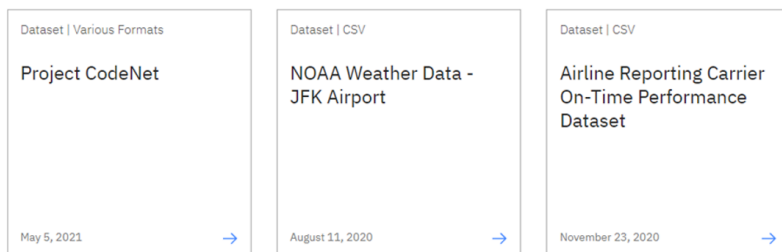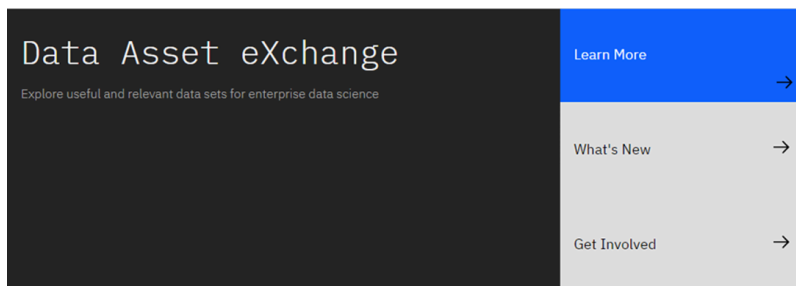
2. In this activity, you will explore **NOAA Weather Data dataset**.

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 eight years.
   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.
   This dataset was used to train the weather forecaster model on MAX (https://developer.ibm.com/exchanges/models/all/max-weather-forecaster/).

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.

   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 targz 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.

5. Most datasets are complemented by Python notebooks that you can use to explore, pre-process, and analyze the data. The notebooks are hosted on Watson Studio, IBM's Data Science platform. Later in this course, you'll learn more about Watson Studio notebooks, how to sign up and how to run notebooks on them.

6. For now, you can preview the dataset and the notebook (or notebooks) by clicking the **Preview the data and notebooks** as shown in the screenshot below.

Now here, you click on the **Notebook Preview** option on top to view the notebook hosted with this dataset. Explore the steps followed in this notebook to clean data before performing data analysis on this dataset.
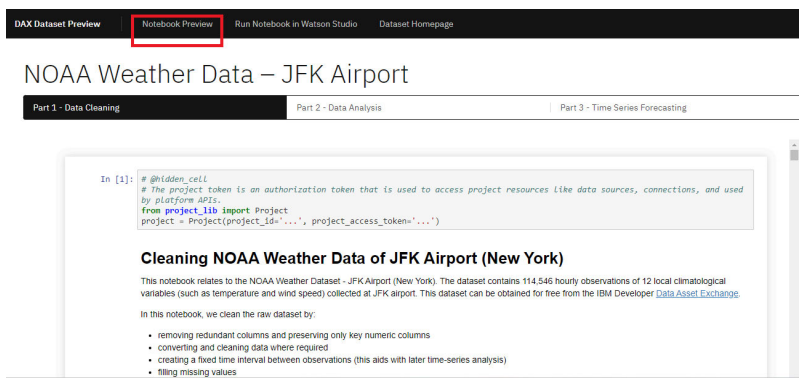
This concludes Exercise 1 of this lab, which introduced the Data Asset Exchange. You may proceed to Exercise 2.

7. [Optional] If you have already registered and are comfortable working with notebooks on Watson Studio, you can open the link using the Run Notebook in Watson Studio option and import it into a project. If you are not acquainted with IBM Watson Studio, you can skip this step. Detailed guidance on signing up and getting started with IBM Watson Studio will be provided in Module 7 of this course.

Exercise 2 - 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.

For more details, please visit - https://github.com/CODAIT/max-central-repo webpage.

The curated list includes deployable models, which you can run as a microservice locally or in the cloud on Docker or Kubernetes, and trainable models where you can use your own data to train the models. Some of the models are already built for you to test. Let's test one of the models.

In this exercise, explore the **Object Detector** model hosted on CodePen platform. This model recognizes the objects present in an image. The model consists of a deep convolutional net base model for image feature extraction, together with additional convolutional layers specialized for the task of object detection, trained on the COCO data set. The input to the model is an image, and the output are extracted objects from the image, appropriately labeled.

CodePen is a social development environment. At its heart, it allows to write code in the browser and see the results of it as you build. It is a useful and liberating online code editor for developers of any skill and is particularly empowering for people learning to code.

1. Navigate to **CodePen** webpage.

2. Select **MAX TFJS models** as shown in the screenshot below. Here the **Image Segmenter**, divides an image into regions or categories that correspond to different objects or parts of objects. Every pixel in an image is allocated to one of a number of these categories.
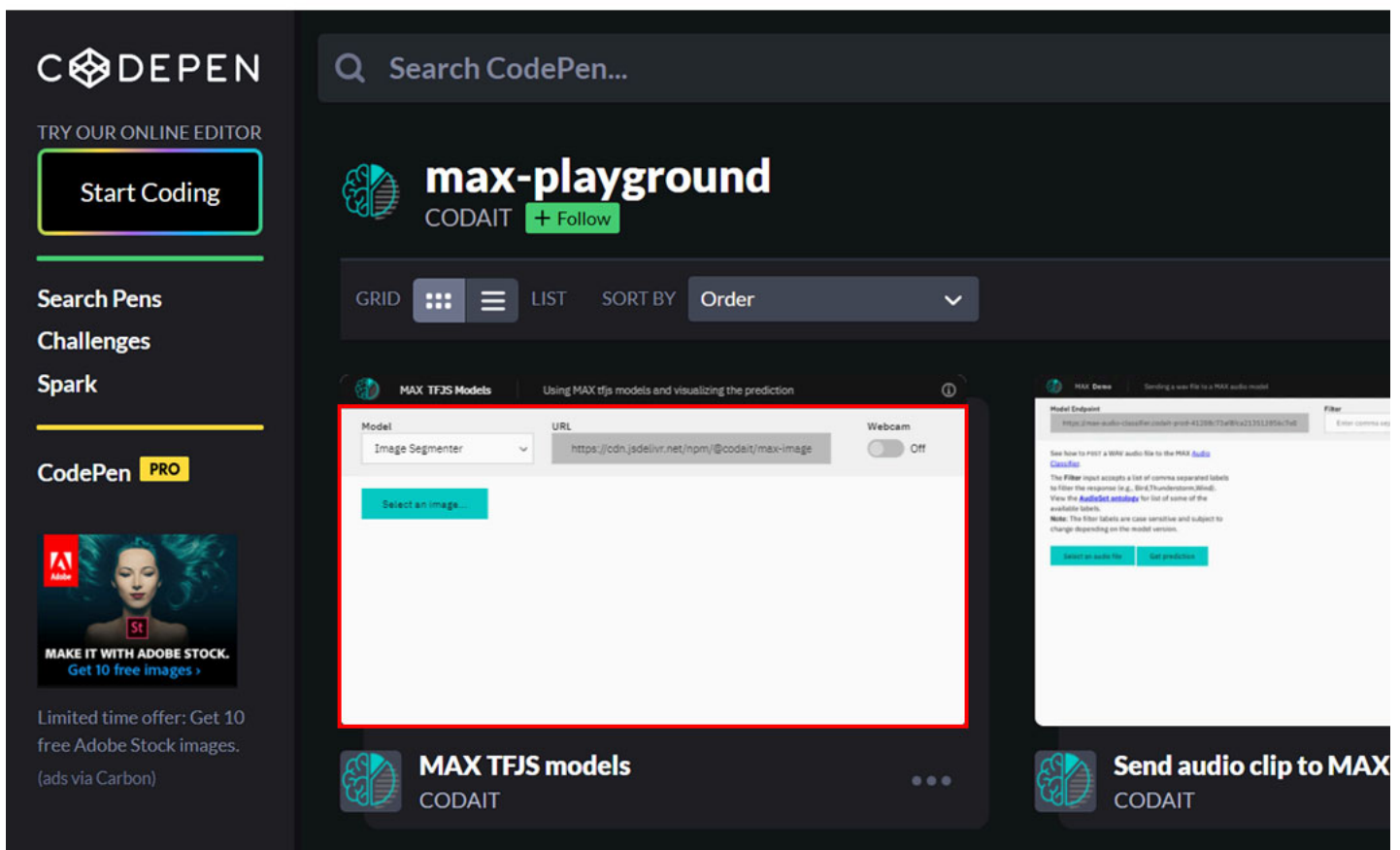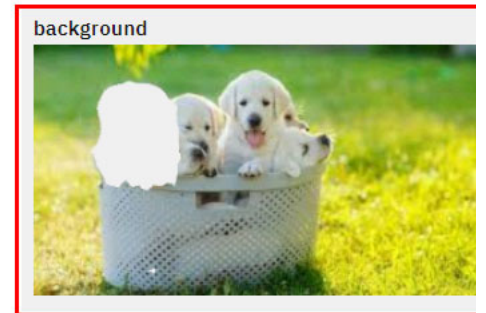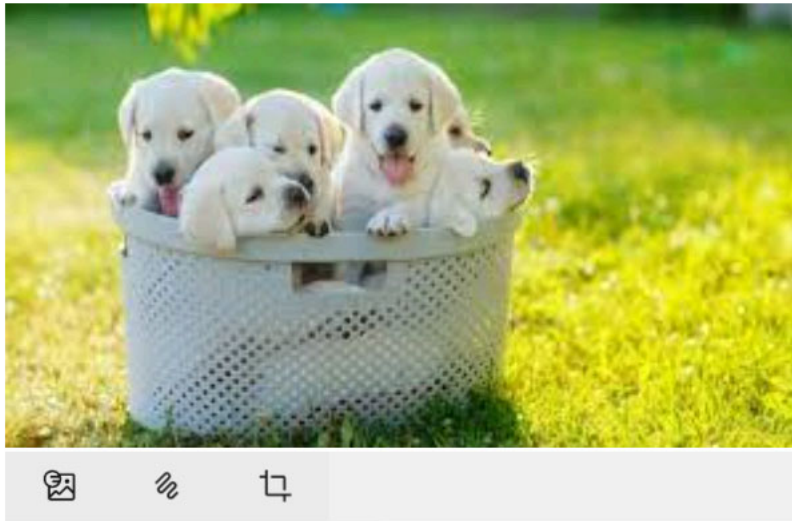

3. Click on **Select Image** and upload an image. You may choose images with a person, dog, cat, truck, car, and so on, which are labels the model has been trained on.
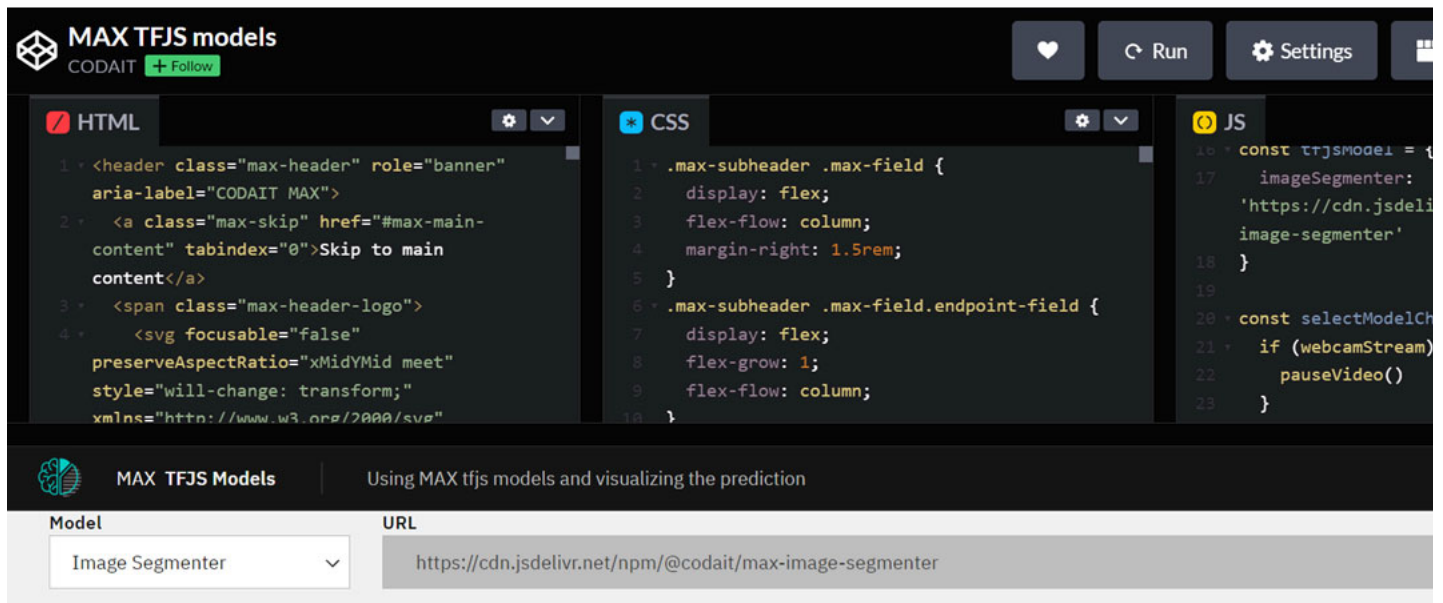
▶ Click here for all the labels the model is trained on

    4. Click the icon **Extract prediction** as shown below:
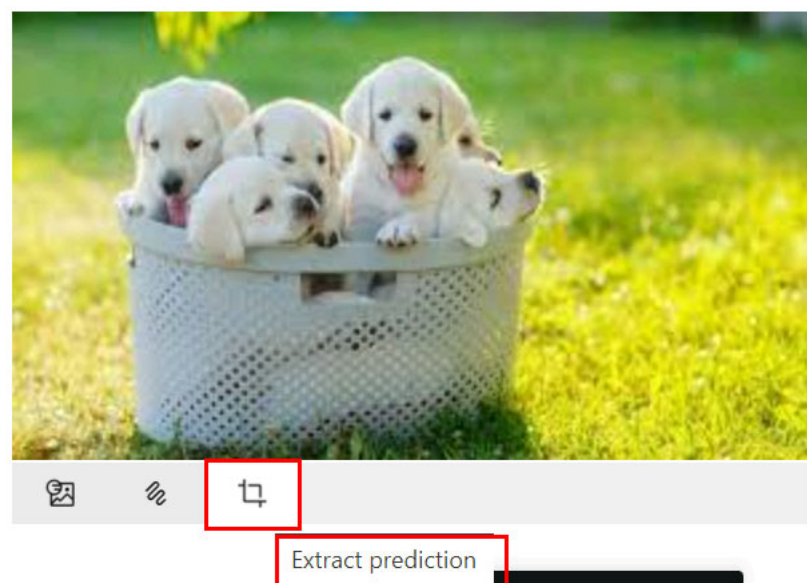
You can see the output of the prediction on the basis of the uploaded image.

Here the background and the dog image are separated, showing two different parts of the image.

You can also try the webcam option, which will show the real-time prediction by the toggle-on webcam option.

This concludes Exercise 2 of this lab, which introduced the Model Asset Exchange (MAX).

You can also watch a demo of the object detector model [here](#).

Author(s)

Joseph Santarcangelo

Other Contributor(s)

Lavanya