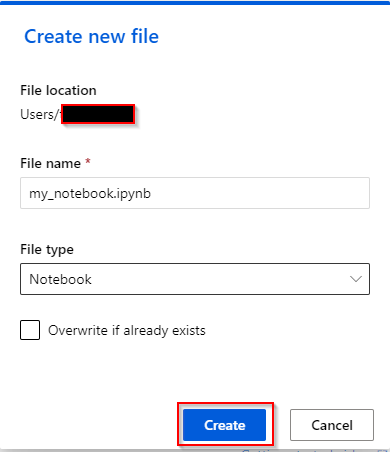
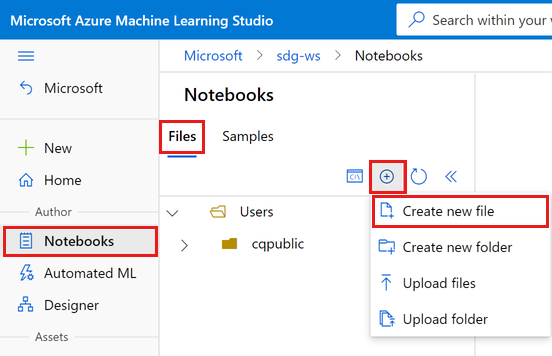
Azure Machine Learning (ML) is a platform for operating machine learning workloads in the cloud. With the Azure ML Studio, you work in a machine learning workspace. A workspace defines the boundary for a set of related machine learning assets. You can use workspaces to group machine learning assets based on projects, deployment environments (for example, test and production), teams, or some other organizing principle.

1. **Notebook features**

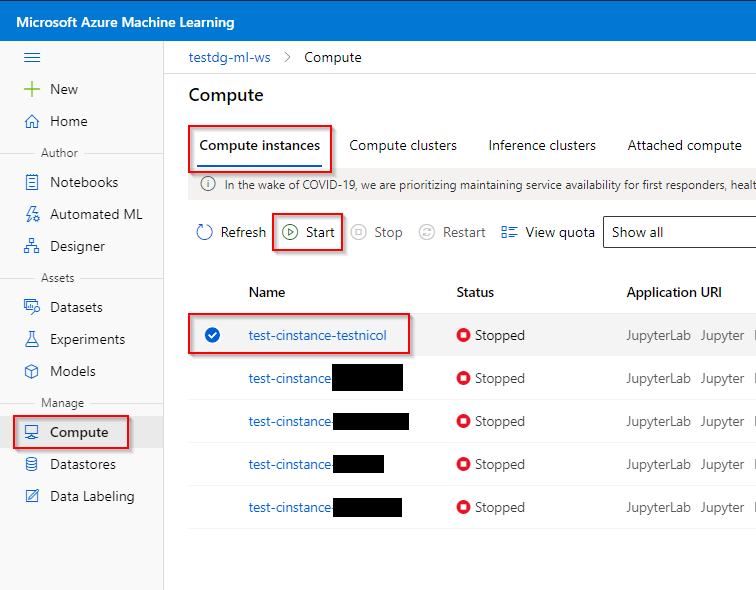
Azure ML Notebooks are Jupyter Notebooks with Python and R kernels integrated with Azure ML Studio. You can run your explorative analysis, pre-processing and post-processing actions on these Jupyter Notebooks using an user-specific compute instance (a personal Virtual Machine).

Although you can train small machine learning models on the user-specific compute instance, when higher computational power is needed, we recommend to use the compute cluster (a scalable compute resource) provided to you and connected to the workspace.

* **Creating a Notebook**



* **Starting/Stopping compute instance**



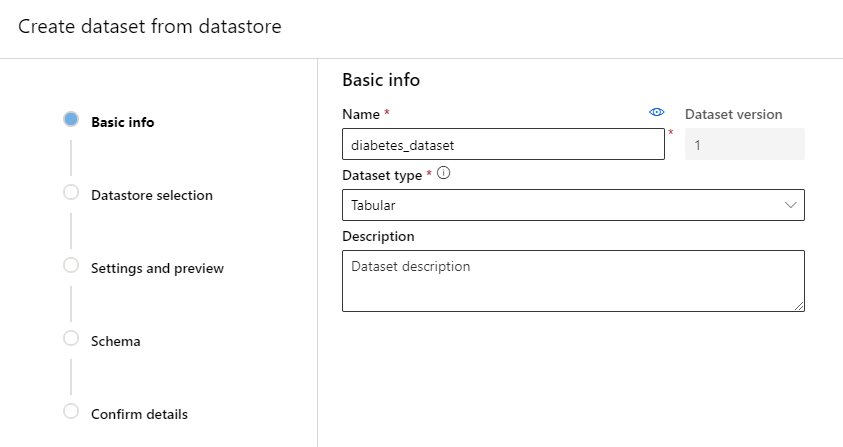
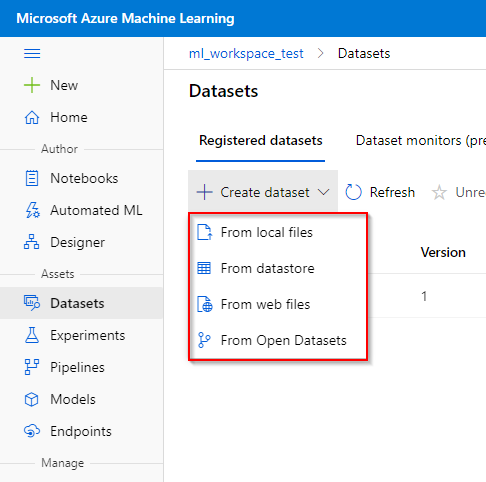
1. **Data provision**

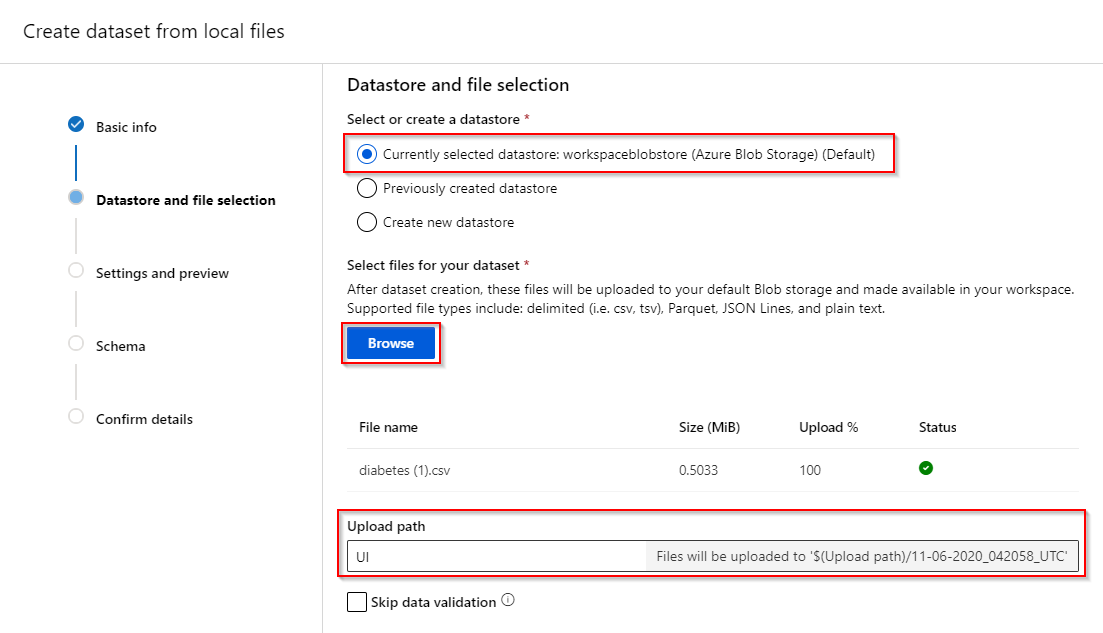
Datasets are versioned packaged data objects that can be easily consumed in experiments. Datasets are the recommended way to work with data, and are the primary mechanism for advanced Azure Machine Learning capabilities like data labeling and data drift monitoring.

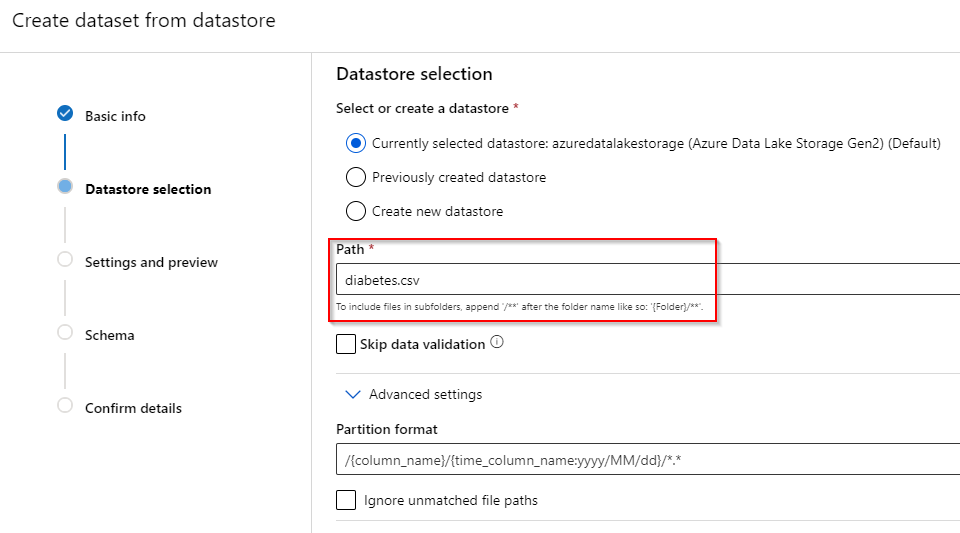
* **Registering a dataset from ( a local file – Data Lake GEN 2 – a SQL Database )**

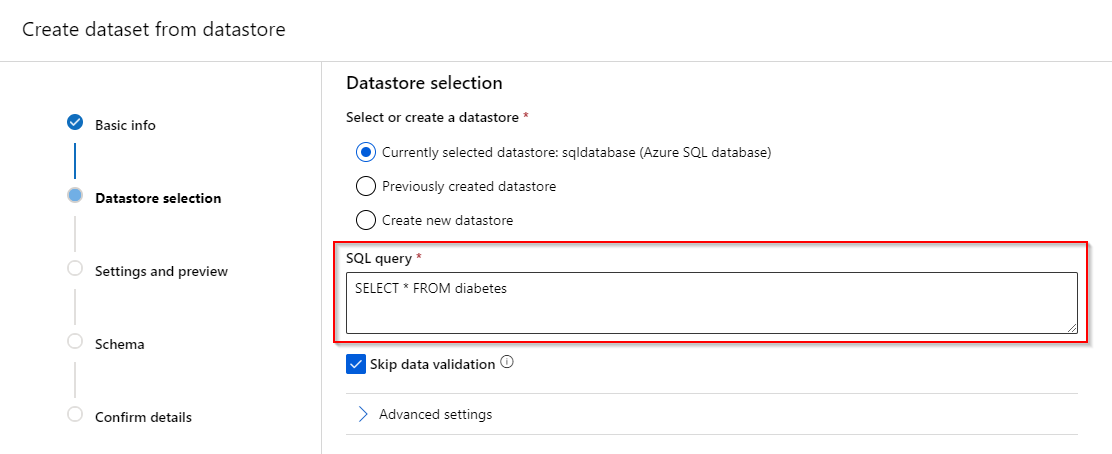
A registered dataset is a dataset that is registered in the workspace of the Azure ML Studio. A registered dataset can be referenced, either graphically or programmatically, when creating an experimental run using a Machine Learning model.

- types of datasets: Tabular (e.g. csv-file) and File (e.g. images)

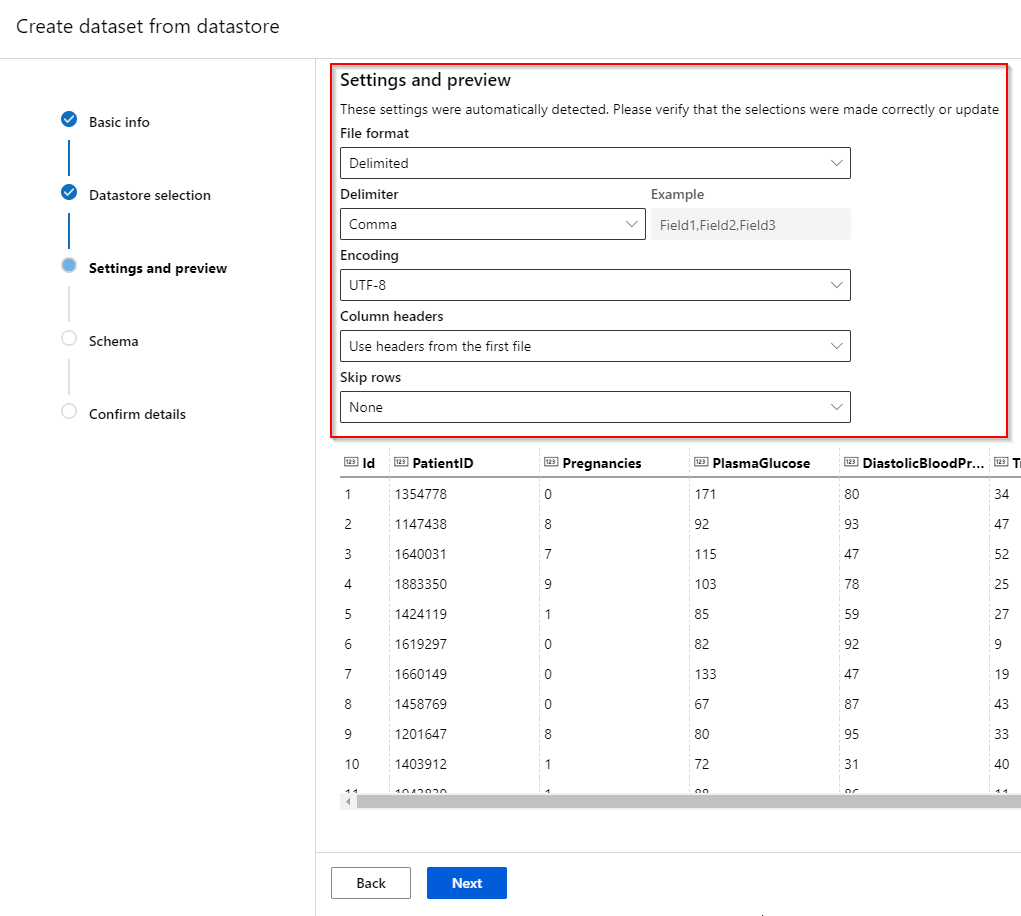


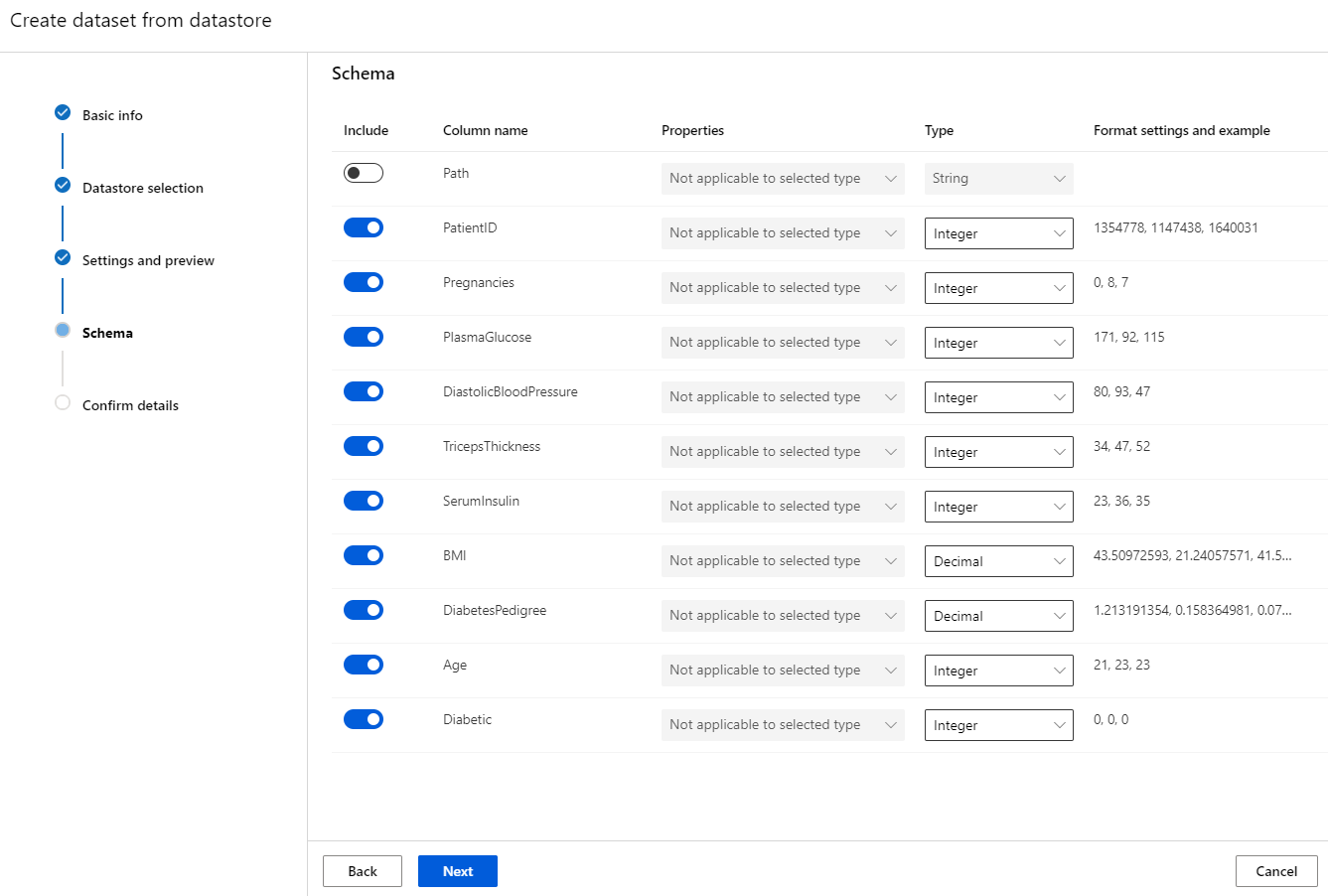


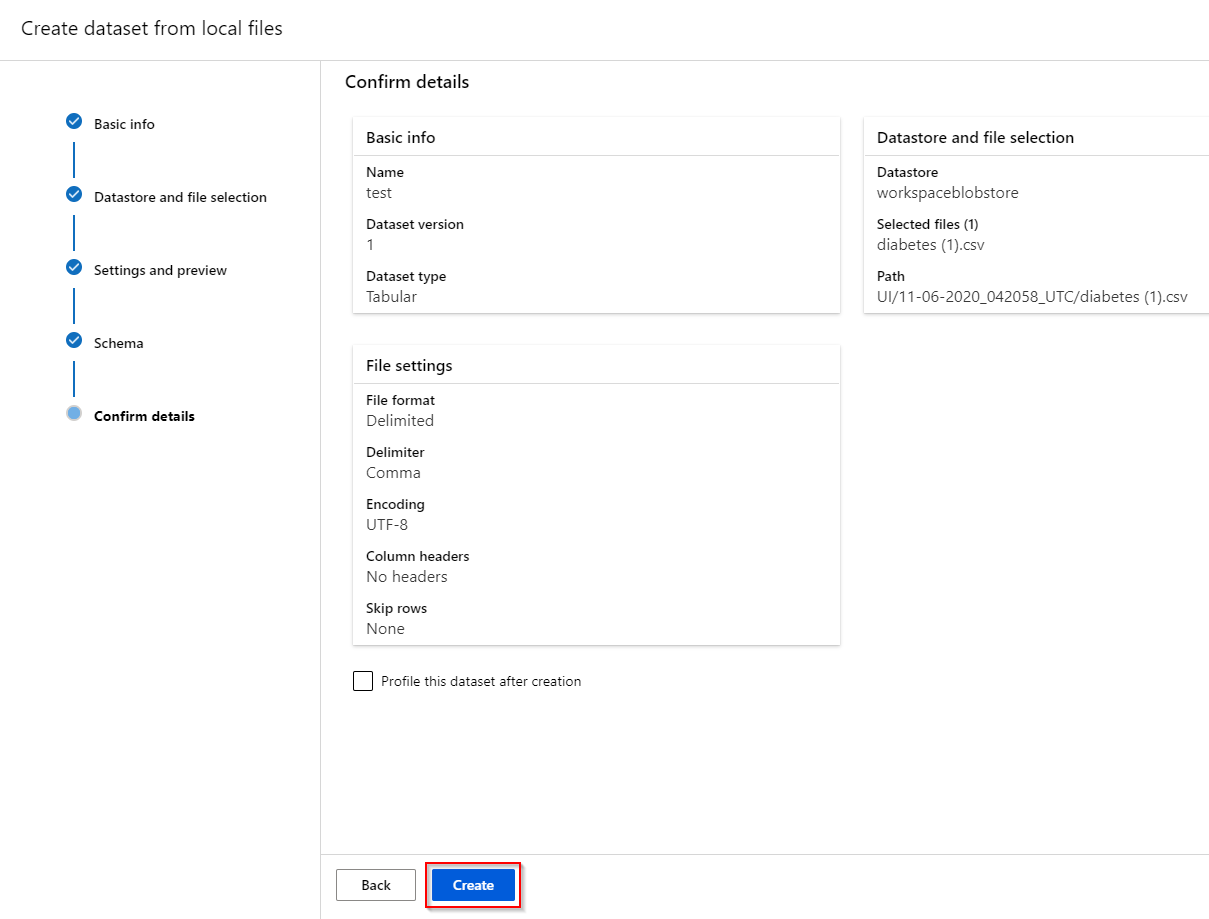




- File format: delimited, plain text, parquet file or JSON Lines

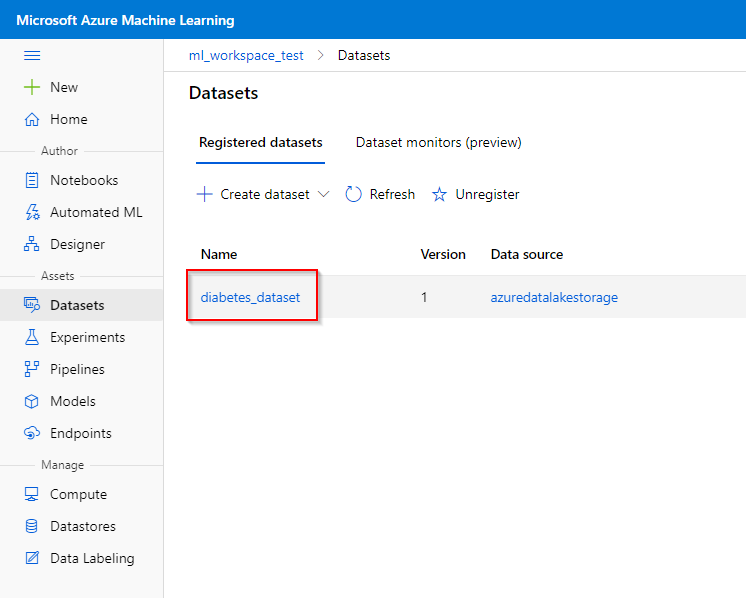
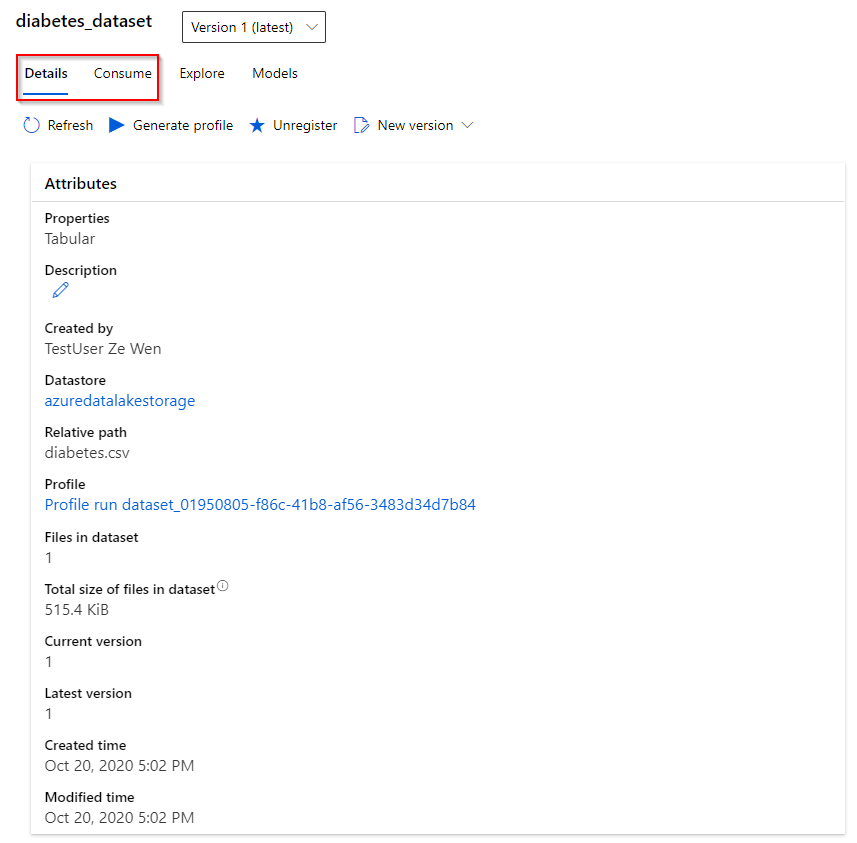




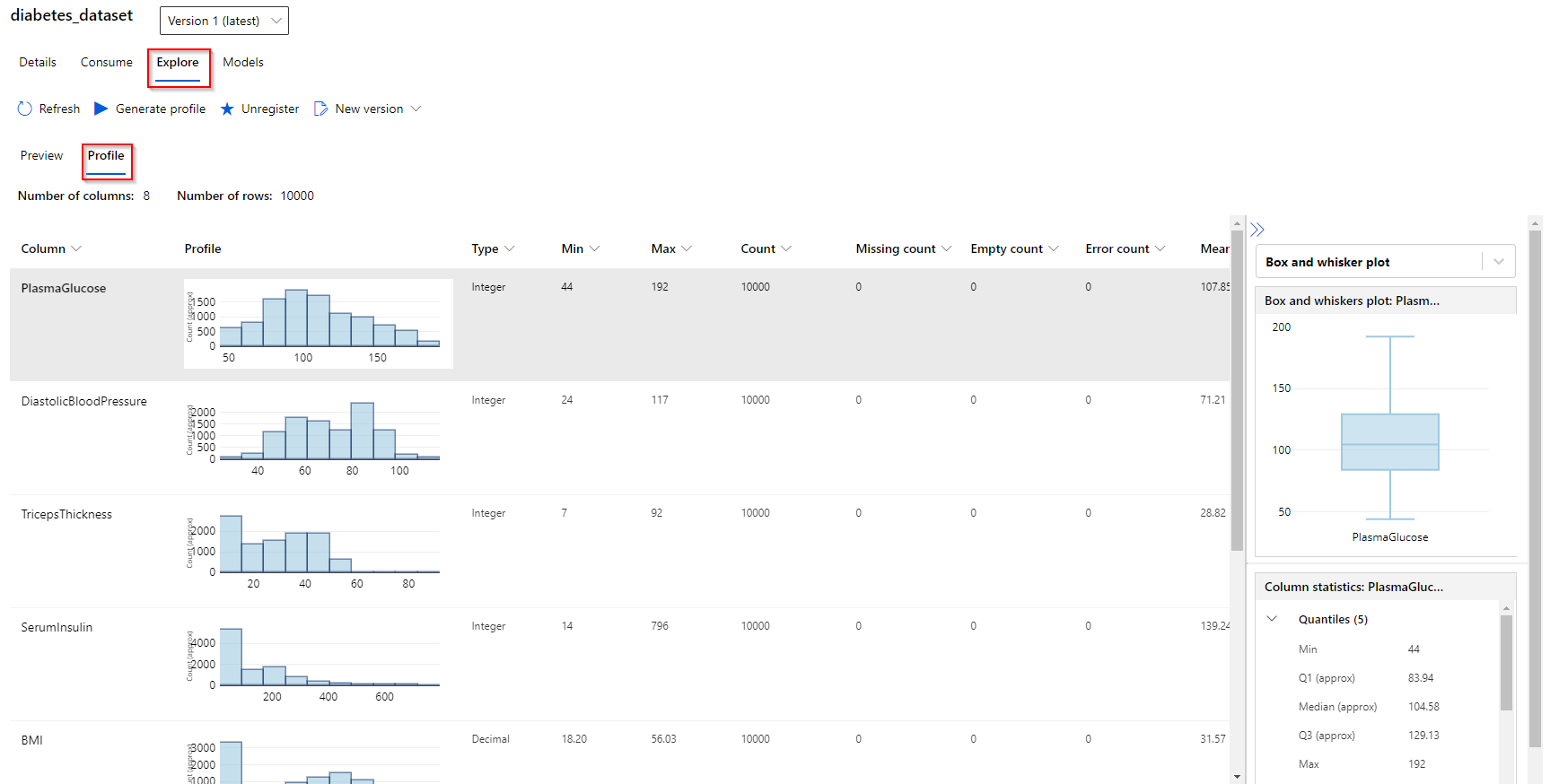


* **Viewing the dataset**

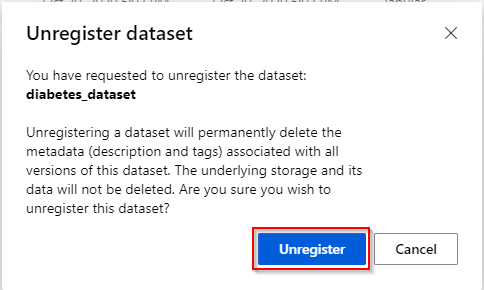
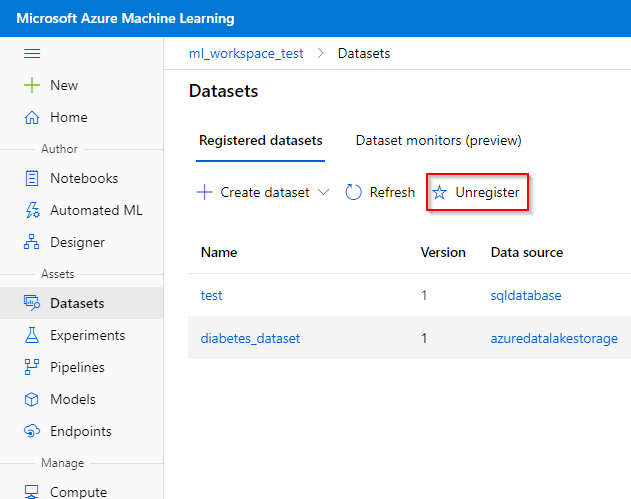
After a dataset is registered, you can obtain information from this dataset in the workspace.

- Simple characteristics of your attributes: histogram distribution, type, min-max value, mean, standard deviation



* **Unregister a dataset**



1. **Model training**

This section provides user instructions on when training a machine learning model by using a registered dataset. There are three ways to construct and train machine learning models in the workspace:

* Training a fast ML model set using the **AutoML feature**.
* Pre-process, train and post-process your data using the drag-and-drop **Designer feature**.
* you can train a custom machine learning model using Python/R script, using the **Notebook feature**.

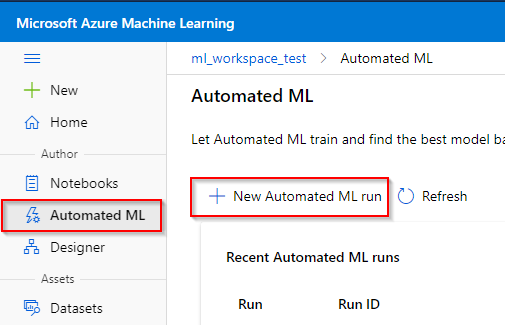
1. **AutoML**

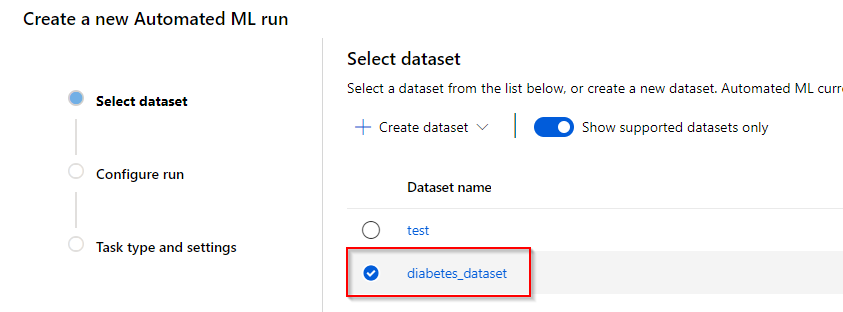
Automated Machine Learning (AutoML) enables you to try multiple algorithms and preprocessing transformations with your data. This, combined with scalable cloud-based compute makes it possible to find the best performing model for your data without the time-consuming manual trial and error that would otherwise be required.

The AutoML feature can handle the following machine learning tasks:

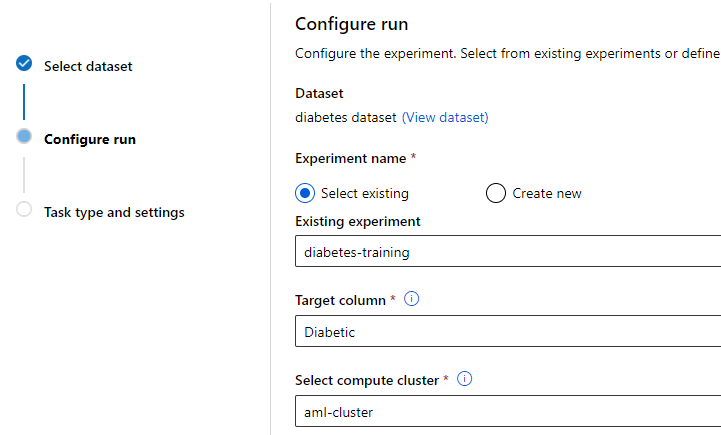
* Classification
* Regression
* Time-series forecasting

AutoML automatically pre-processes the provided data (scaling and normalization). The AutoML feature currently only supports TabularDataset.



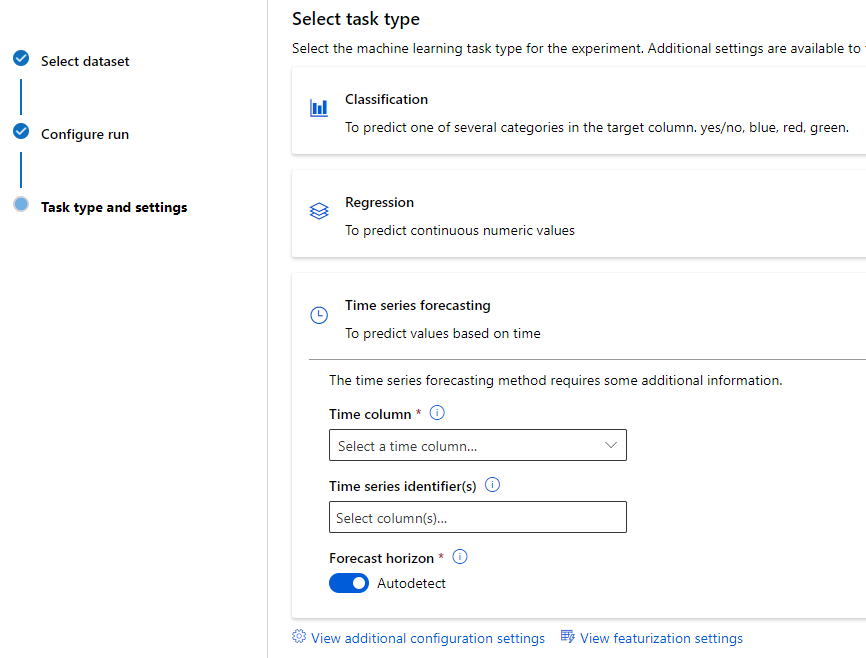


- target column: the column in the dataset that you want the model to predict



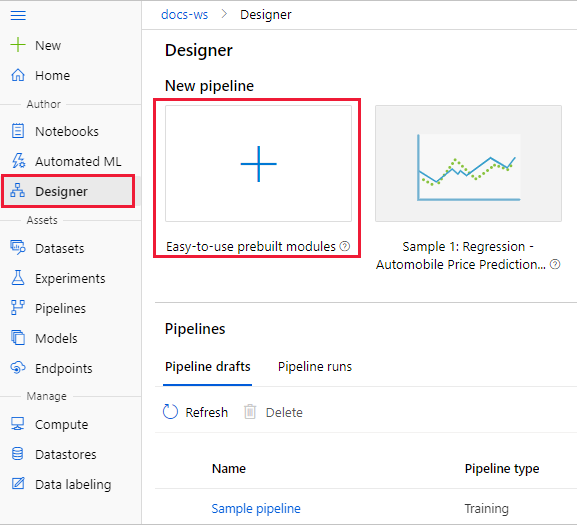
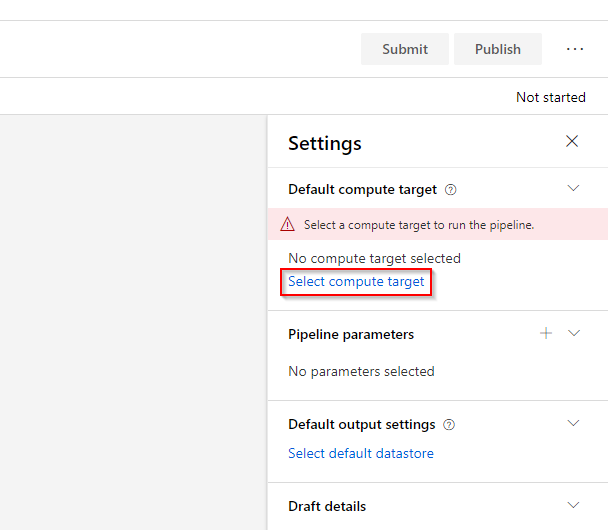
- In the classification category, you can specify deep learning capabilities of AutoML. In the time-series forecasting category, additional information such as time column and forecast horizon need to be provided.

- Additional configuration settings can be provided at the bottom, such as primary metric to train, validation type, exit criteria, etc. Finally, you can also specify to include feature engineering capabilities.

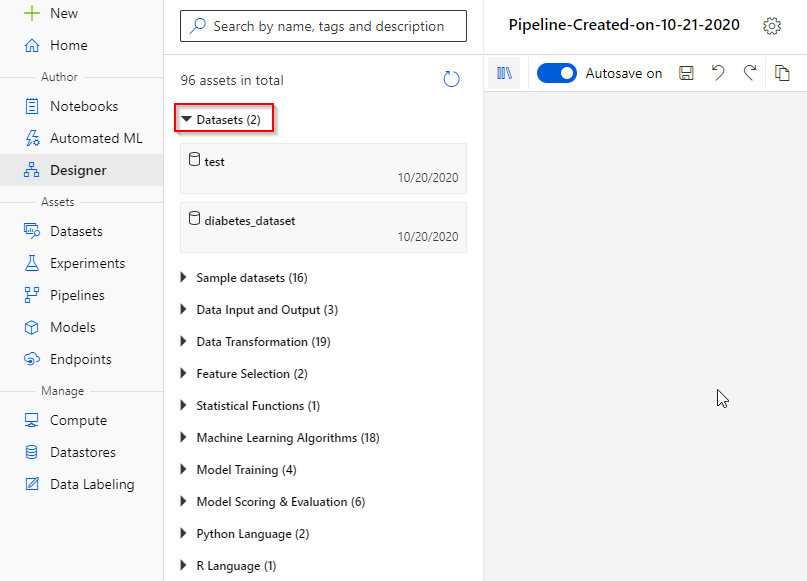


1. **Designer**

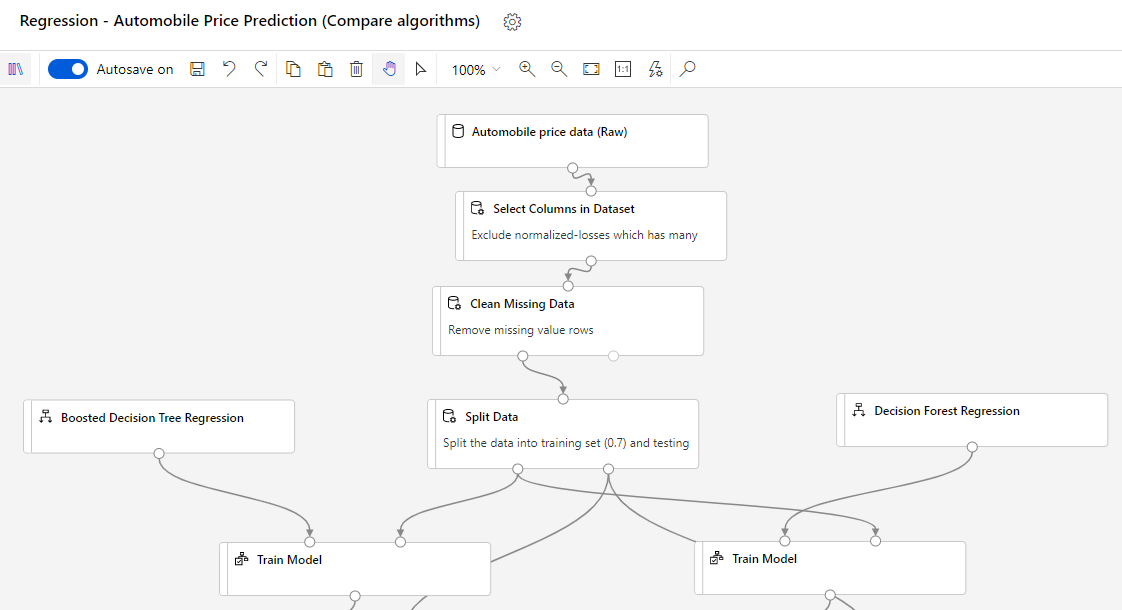
In an enterprise data science process, you'll generally want to separate the overall process into individual tasks, and orchestrate these tasks as pipelines of connected steps. Azure Machine Learning designer lets you visually connect datasets and modules on an interactive canvas to create machine learning models.

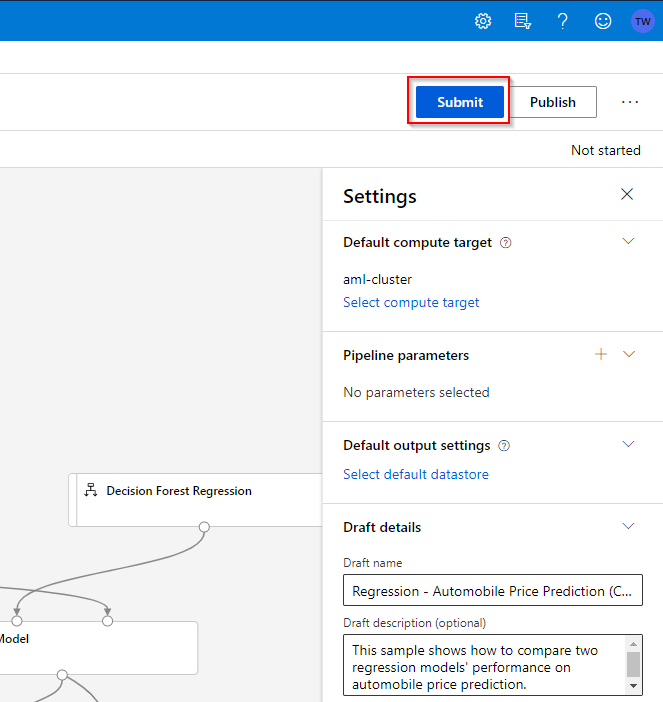
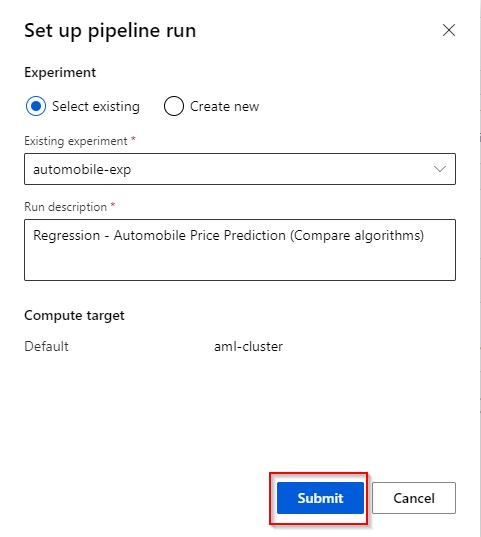
 

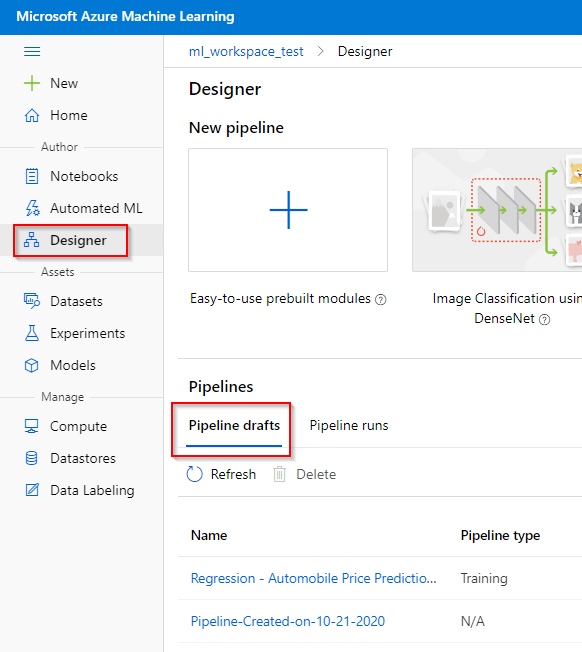
- the different assets/building blocks you can use to pre-process, train and post-process your dataset. In the “Datasets” asset, you will see all the registered datasets in the workspace.



- the Automobile Price Prediction template, which compares the results of two different, trained machine learning models on the Automobile price data.

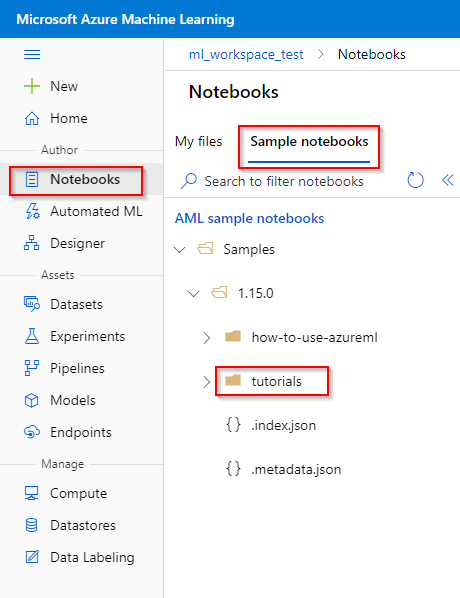




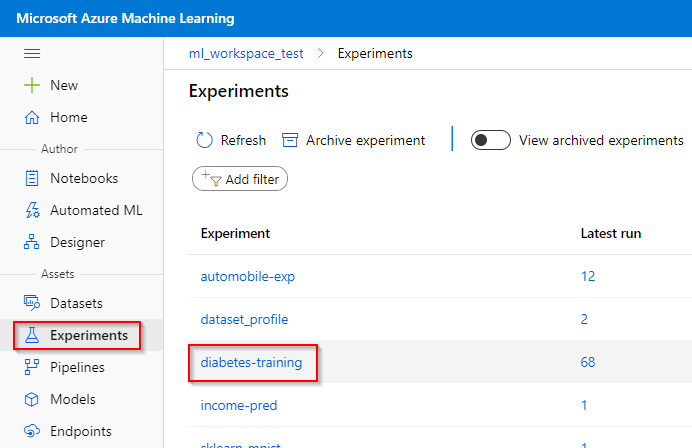
1. **Notebooks**

you can also write and run your own code in managed Jupyter Notebook servers that are directly integrated in the workspace. Microsoft has provided several Machine Learning tutorials on how to connect to the workspace resources and create and run experiments in a programmatic way.

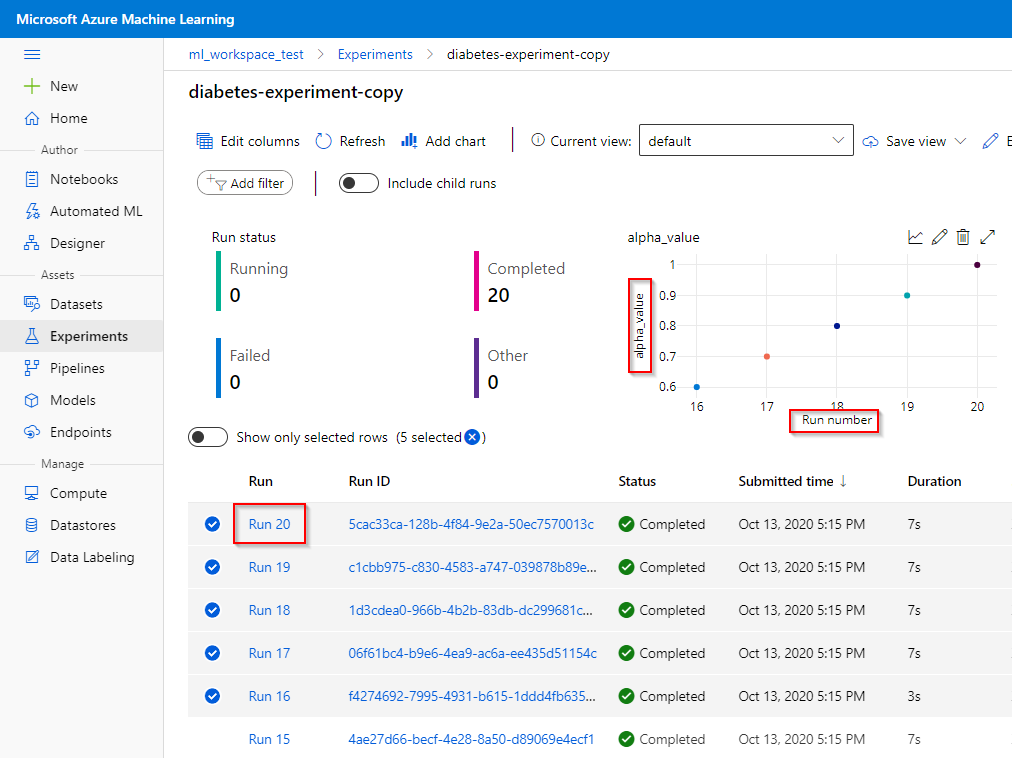


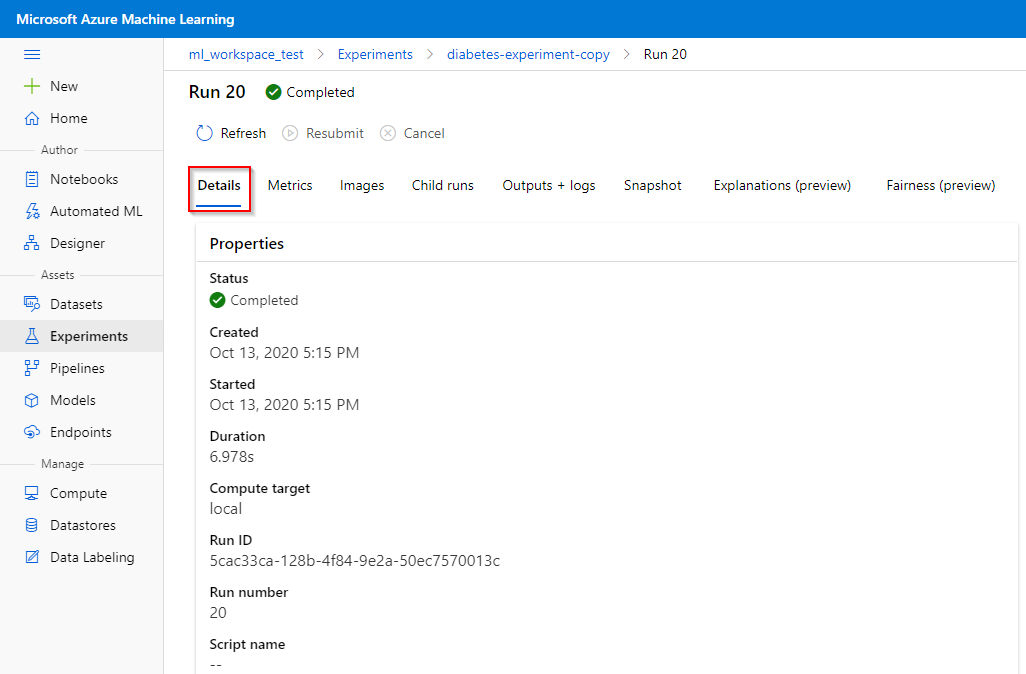
* **Exploring the Experimental runs**

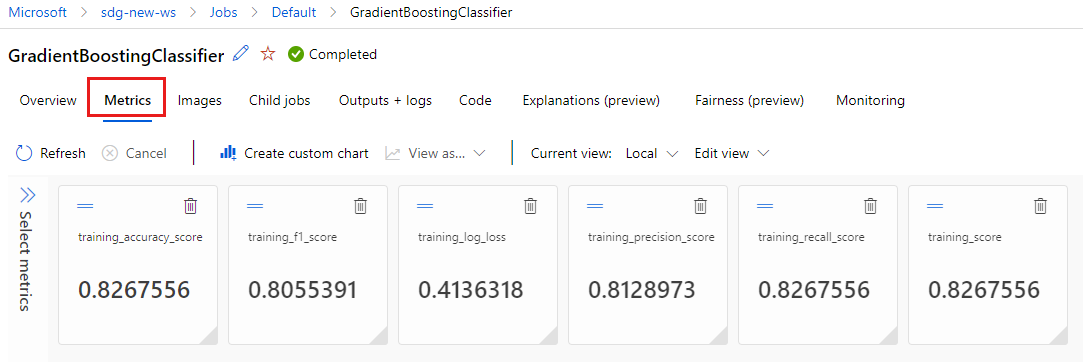
The information on an Azure ML experimental run can be viewed in the “Experiment” tab under the “Assets” category.

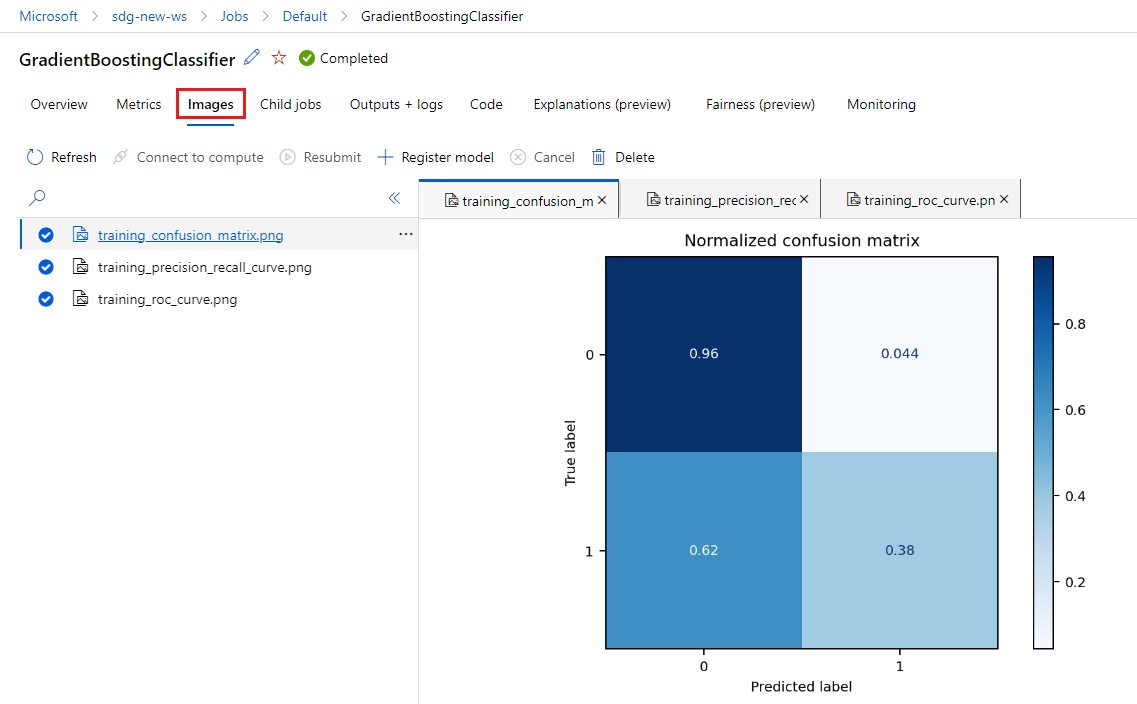


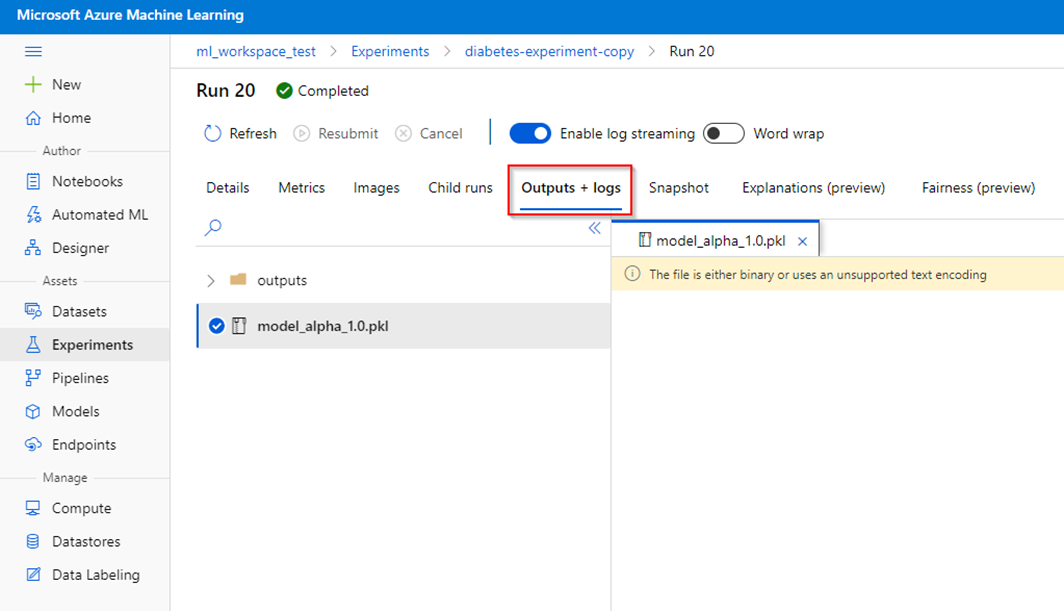
In the Azure ML experiments, you can see which runs are still running, failed or completed. If you assigned any custom, logged metrics to your runs (using the Notebook feature), the evolution of these metrics over different runs can be viewed here.







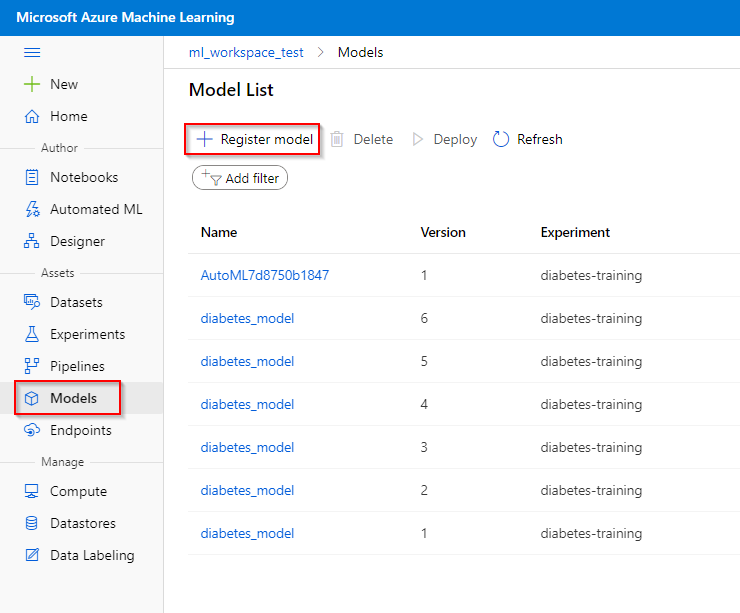




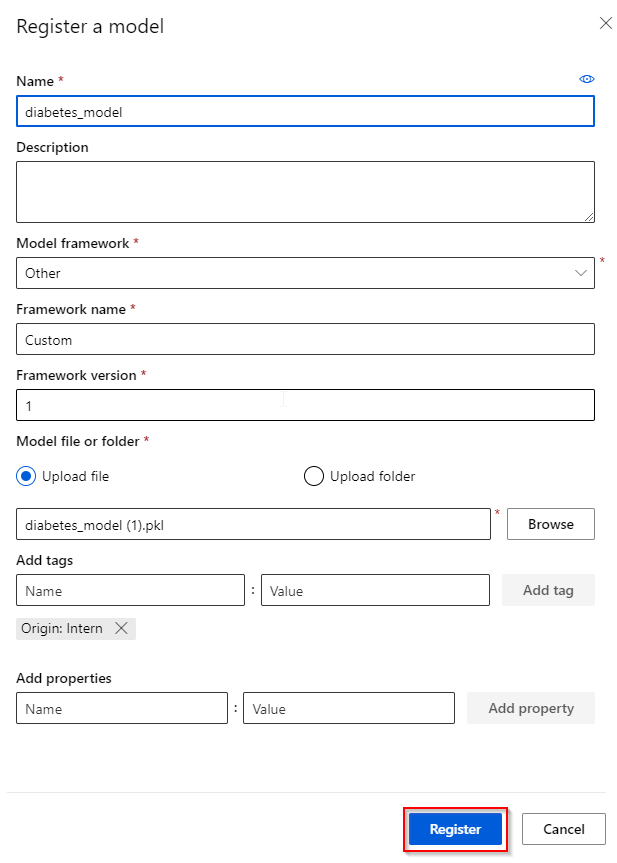
1. **Model registration**

After training the model, you can register it in the workspace from the trained machine learning model runs. Registering a model might be beneficial to manage the machine learning environment of your organization, e.g., be able to reference the model in the Notebook feature.

1. **Registering a model from registered experimental runs**
2. **Registering a model from a local model file**

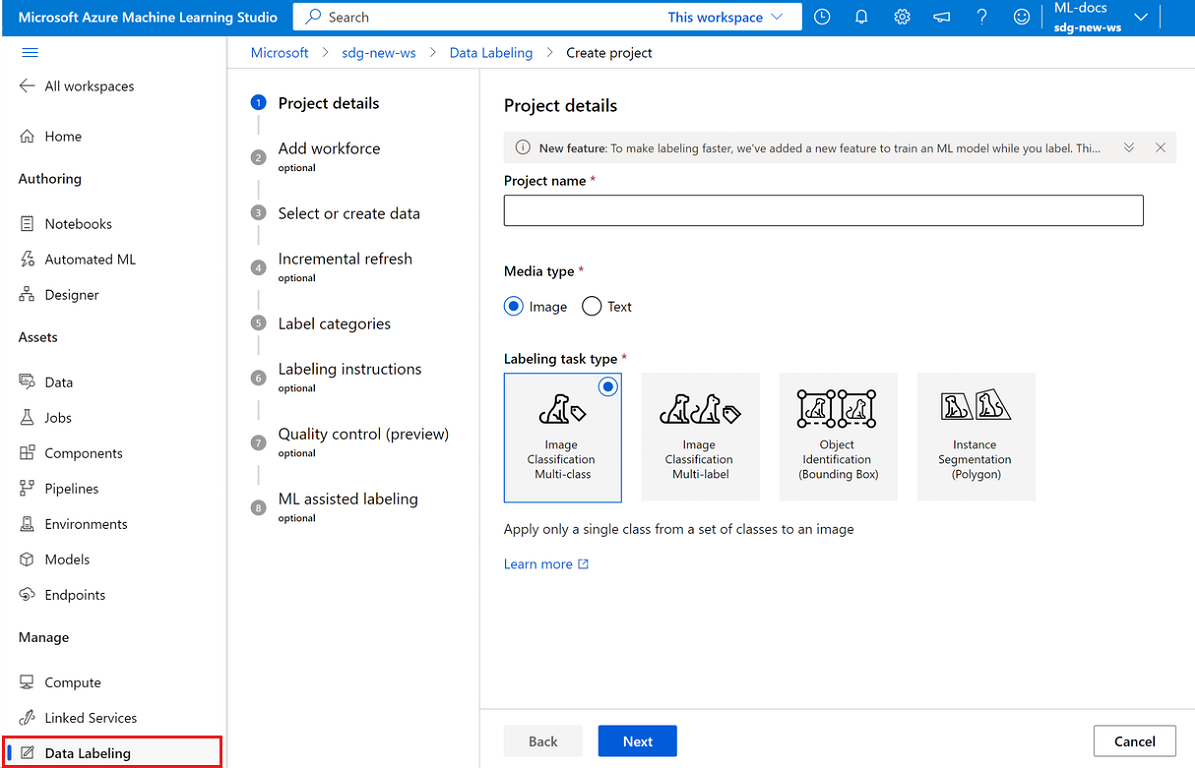


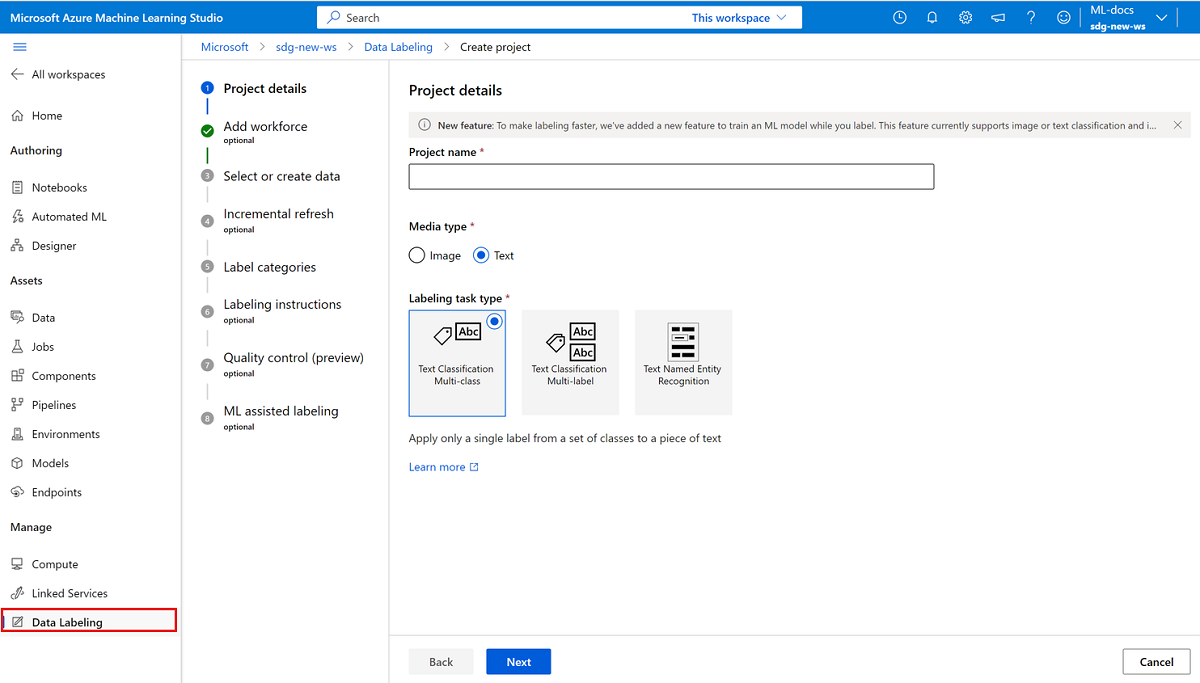
**-** Giving a name to your model and specify the model framework. Also, provide the framework version (for keeping track of your models) and model file corresponding to the model framework given (e.g. the pkl-file). You can also provide tags and properties that are associated with your model.

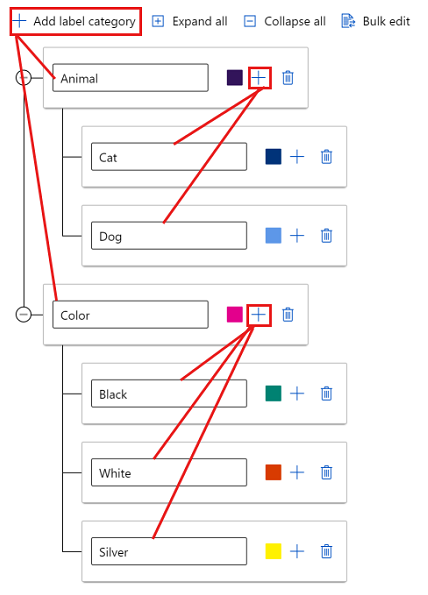
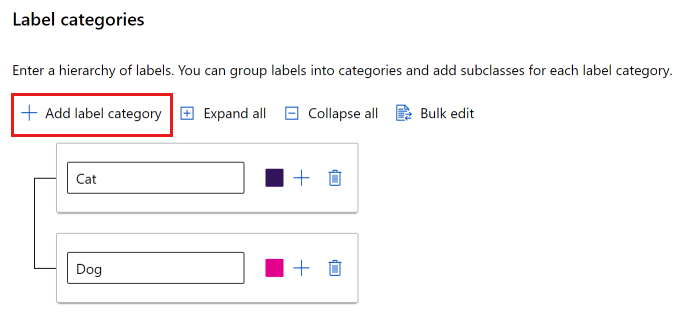


1. **Data Labeling**

Azure Machine Learning data labeling gives you a central place to create, manage, and monitor labeling projects. Use it to coordinate data, labels, and team members to efficiently manage labeling tasks. Machine Learning supports image classification, either multi-label or multi-class, and object identification with bounded boxes.





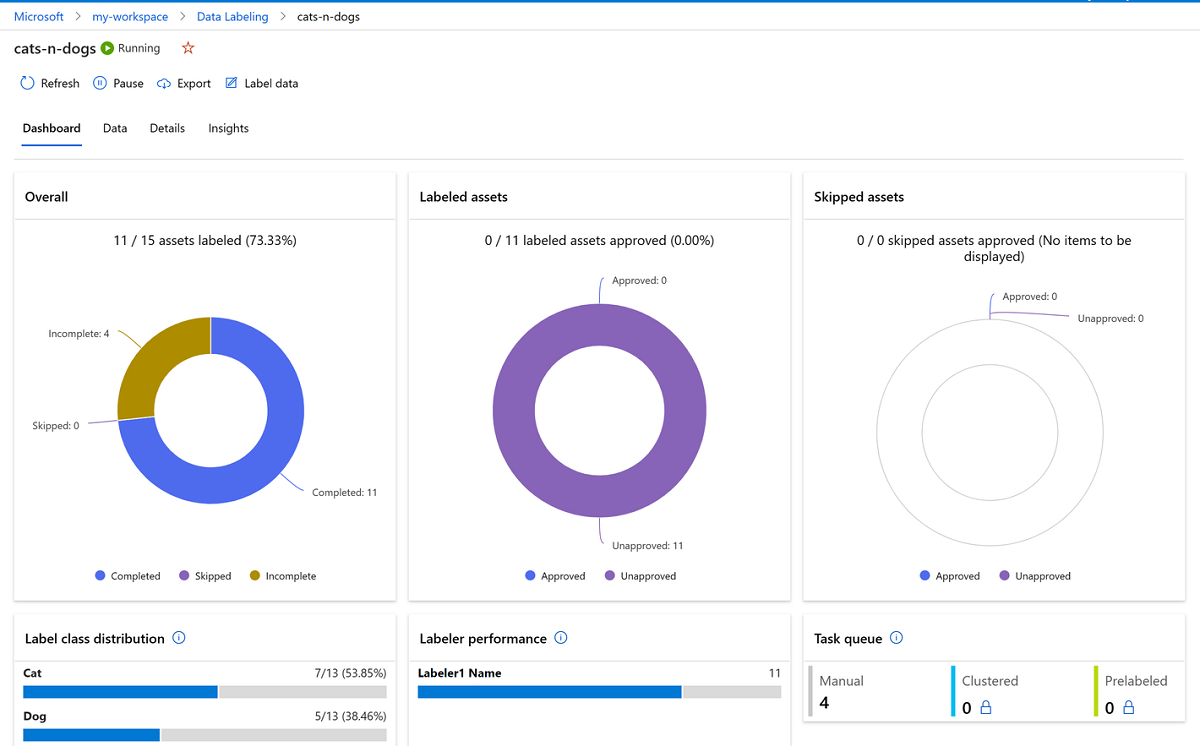


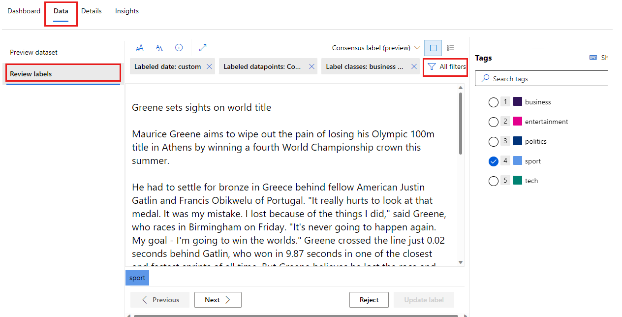
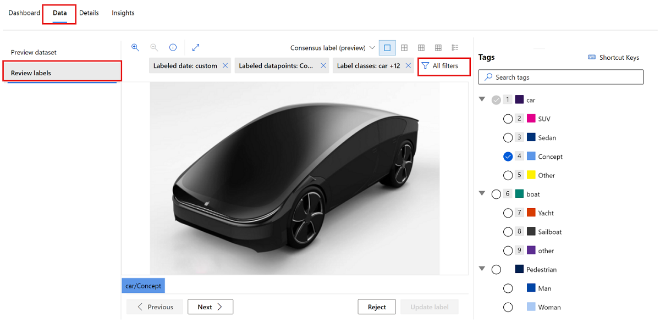
- To accelerate labeling tasks, on the ML assisted labeling page, you can trigger automatic machine learning models. For text, Machine learning (ML)-assisted labeling can handle both file (.txt) and tabular (.csv) text data inputs.

ML-assisted labeling consists of two phases:

* + Clustering
  + Pre-labeling

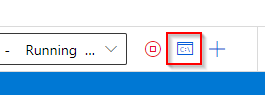
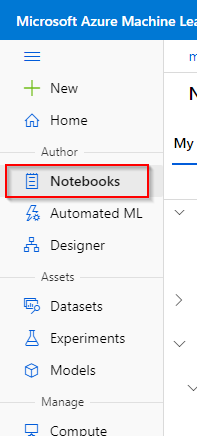
- The clustering phase doesn't appear for object detection models or text classification.





1. **Git integration**

Azure Machine Learning fully supports Git repositories for tracking work - you can clone repositories directly onto your shared workspace file system using Git on your local workstation.

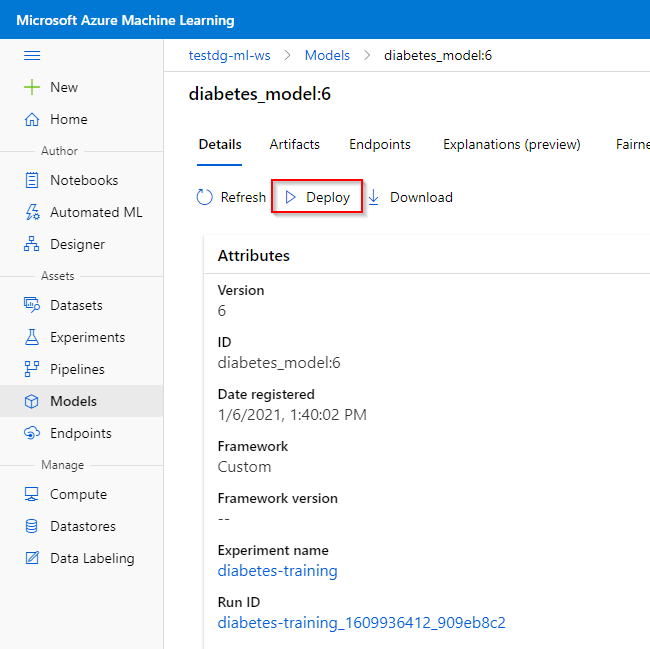
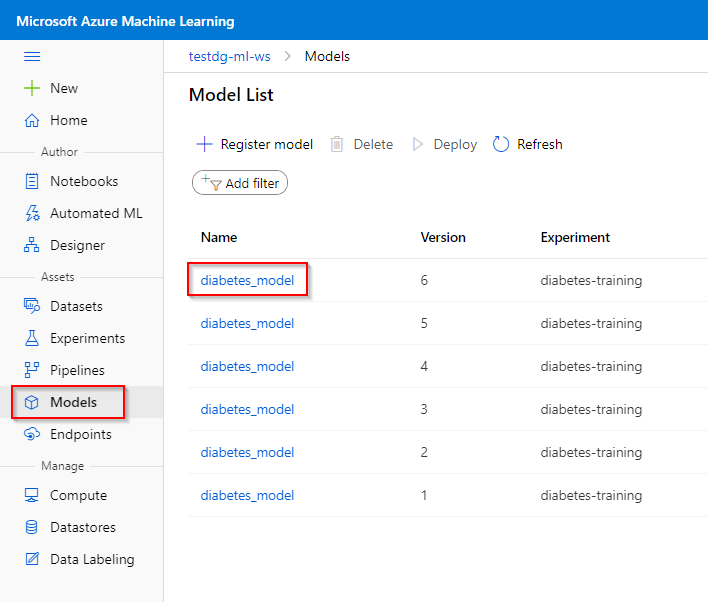


1. **Model Deployment**

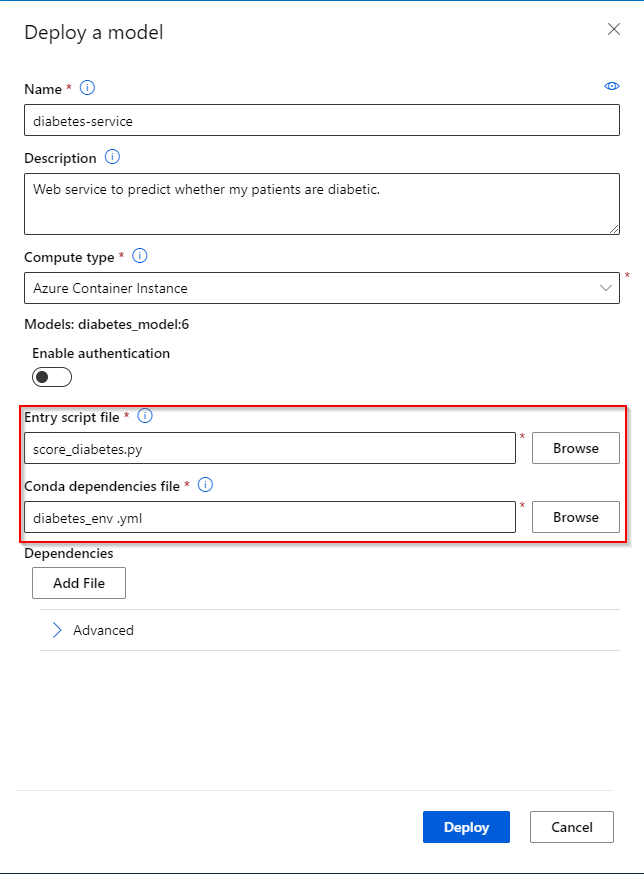
Azure ML Studio has also the feature to deploy a trained machine learning model using a real-time Azure Container Instances (ACIs), a Web Service.

1. **Deploying a web service**

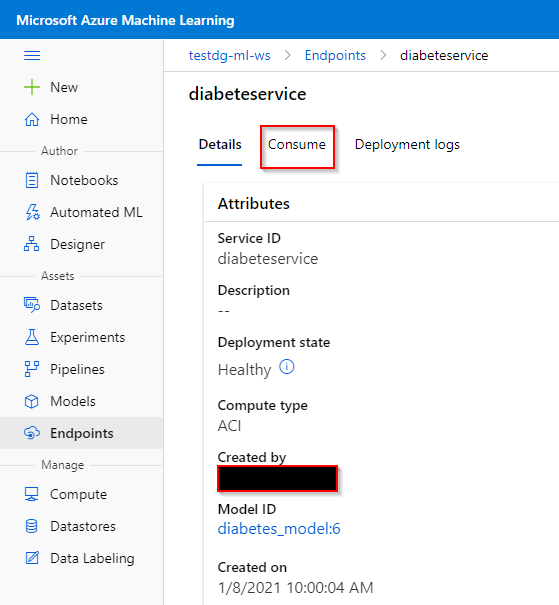
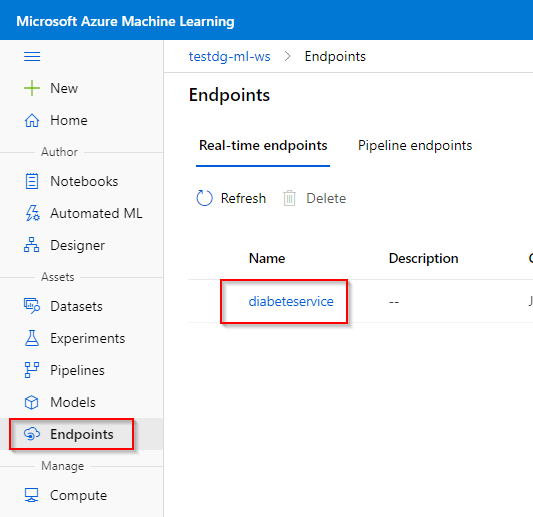
A model can only be deployed after you have registered it in the Azure ML Studio. To deploy a model.

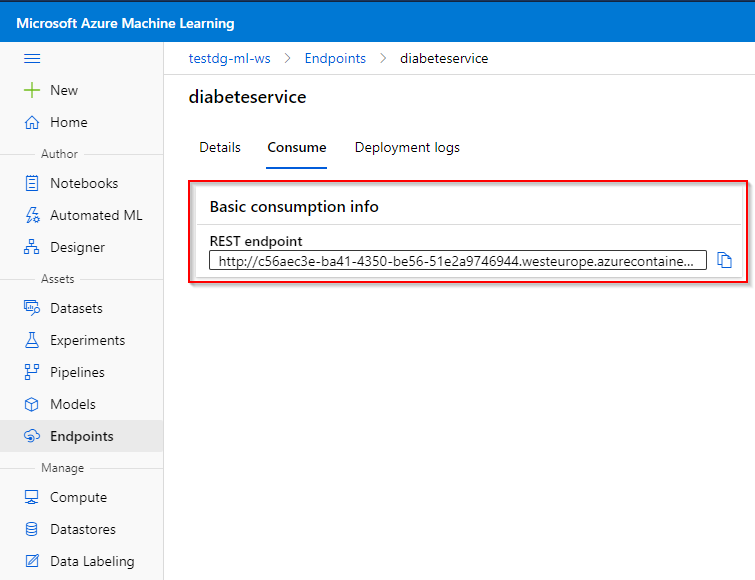


- Provide the entry script and a conda dependencies file. The entry script is used to process the data send to the Web Service. It can be a python file containing an init() and a run(data) function. The conda dependencies files defines the environment used to run the deployed model.



1. **view a web service**





1. **Deleting a web service**

