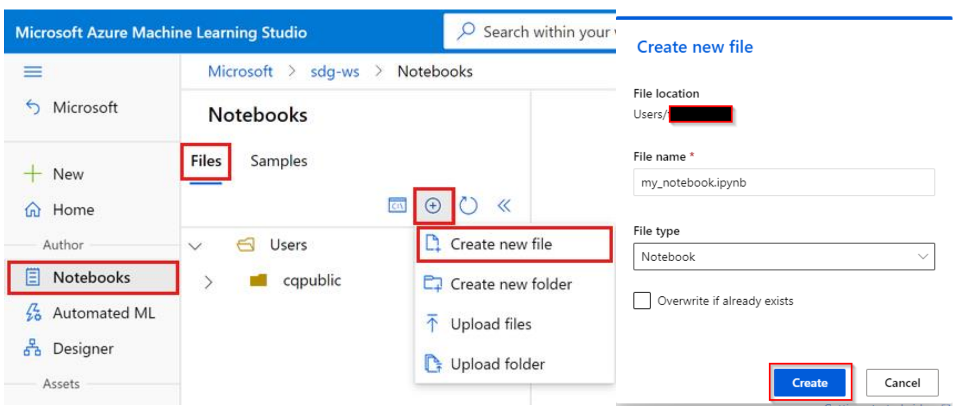
Azure Machine Learning (ML) is a platform for operating machine learning workloads in the cloud. With the Azure ML Studio, you work in a machine learning workspace. A workspace defines the boundary for a set of related machine learning assets. You can use workspaces to group machine learning assets based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

1. Notebook features

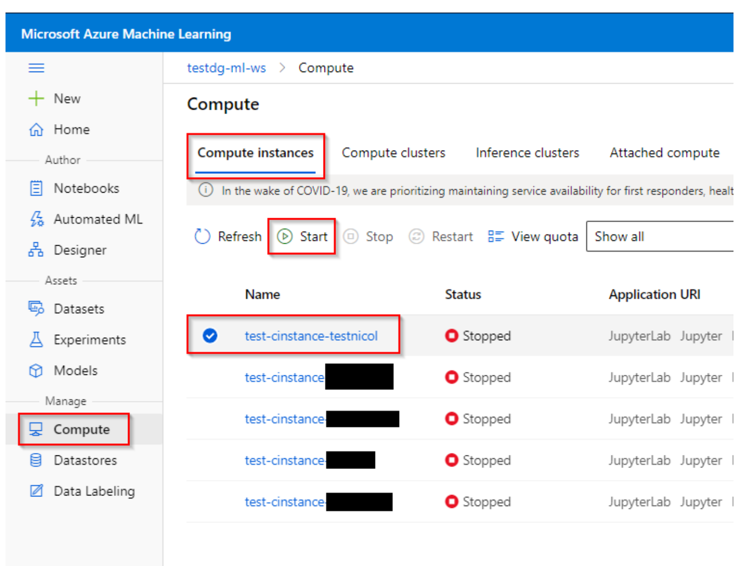
Azure ML Notebooks are Jupyter Notebooks with Python and R kernels integrated with Azure ML Studio. You can run your explorative analysis, pre-processing and post-processing actions on these Jupyter Notebooks using an user-specific compute instance (a personal Virtual Machine).

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**



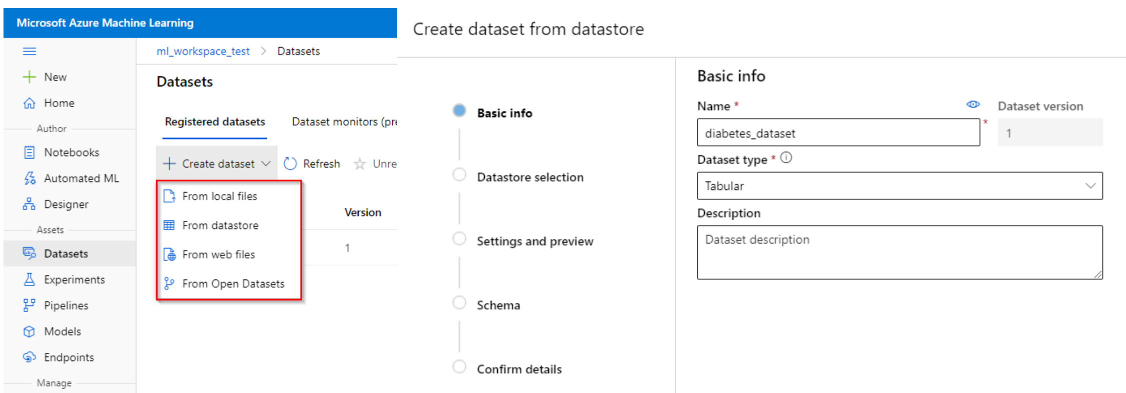
2. Data provision

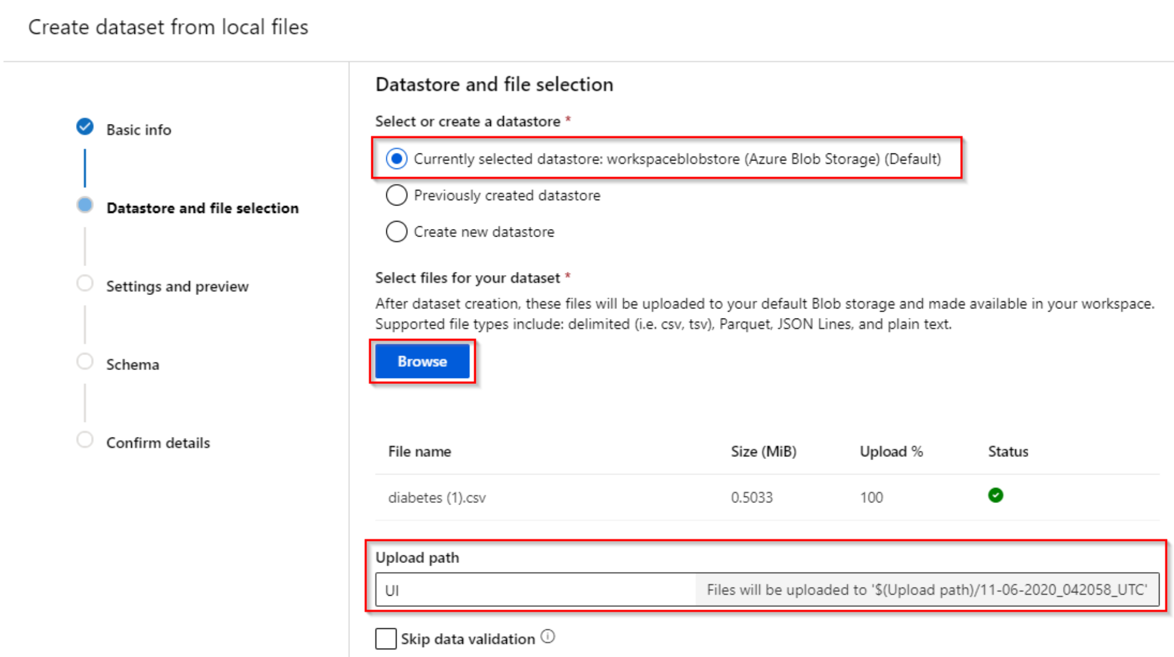
Datasets are versioned packaged data objects that can be easily consumed in experiments. Datasets are the recommended way to work with data, and are the primary mechanism for advanced Azure Machine Learning capabilities like data labeling and data drift monitoring.

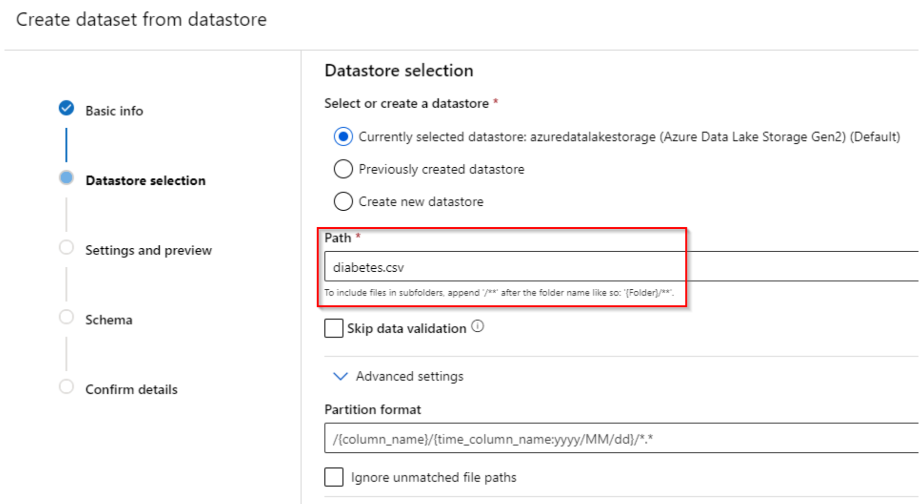
- **Registering a dataset from (a local file – Data Lake GEN 2 – a SQL Database)**

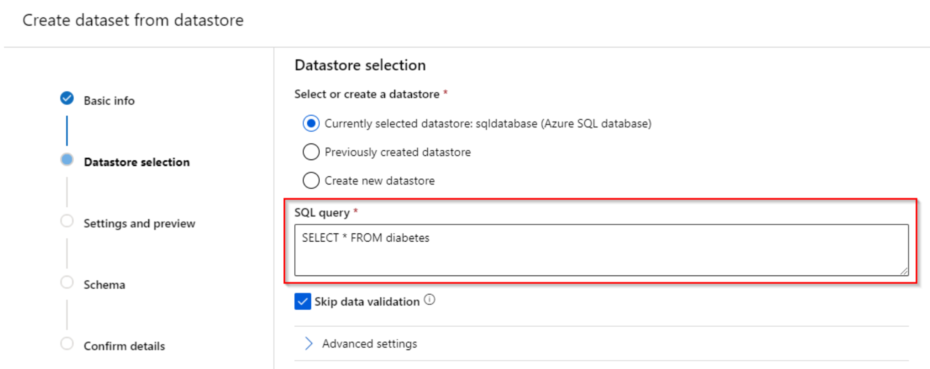
A registered dataset is a dataset that is registered in the workspace of the Azure ML Studio. A registered dataset can be referenced, either graphically or programmatically, when creating an experimental run using a Machine Learning model.

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

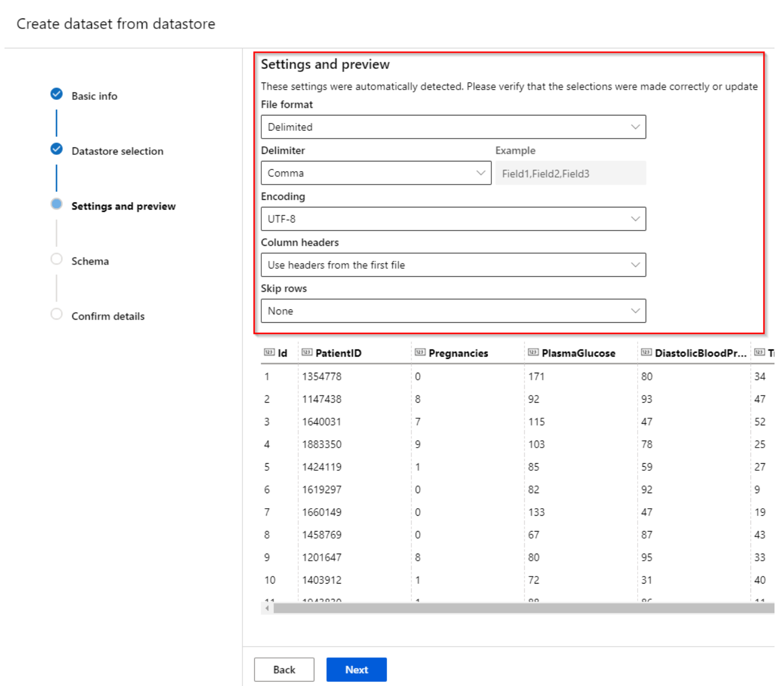


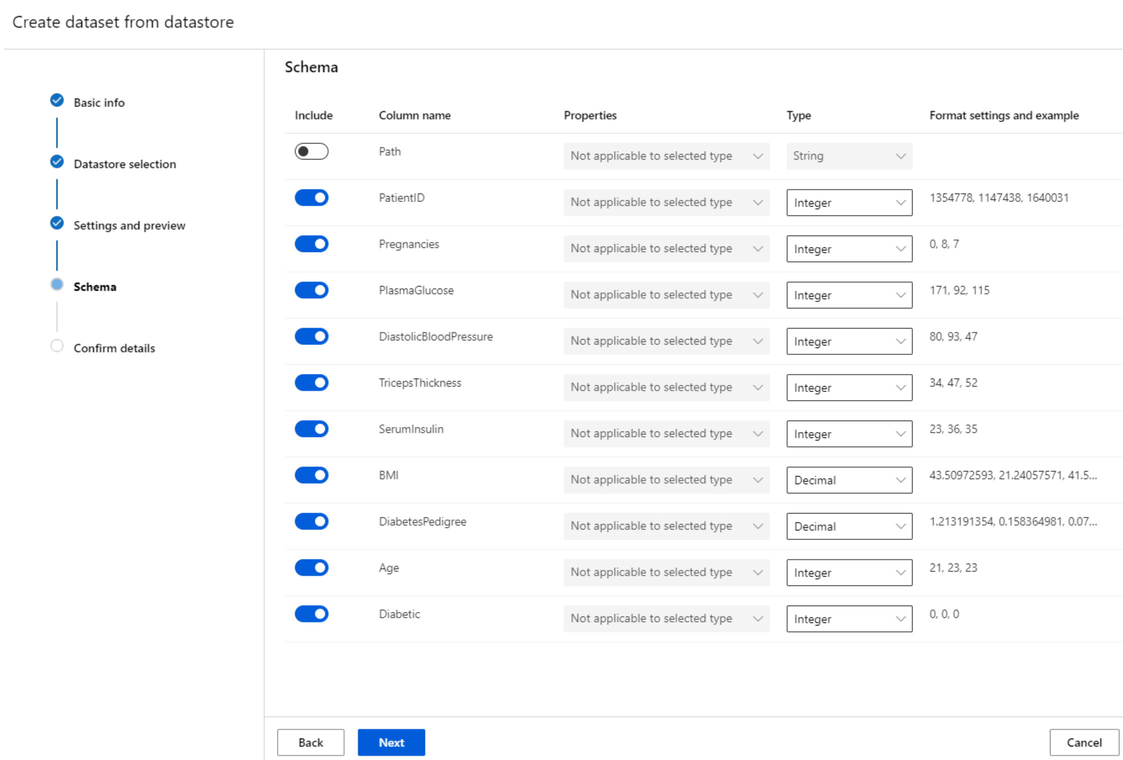


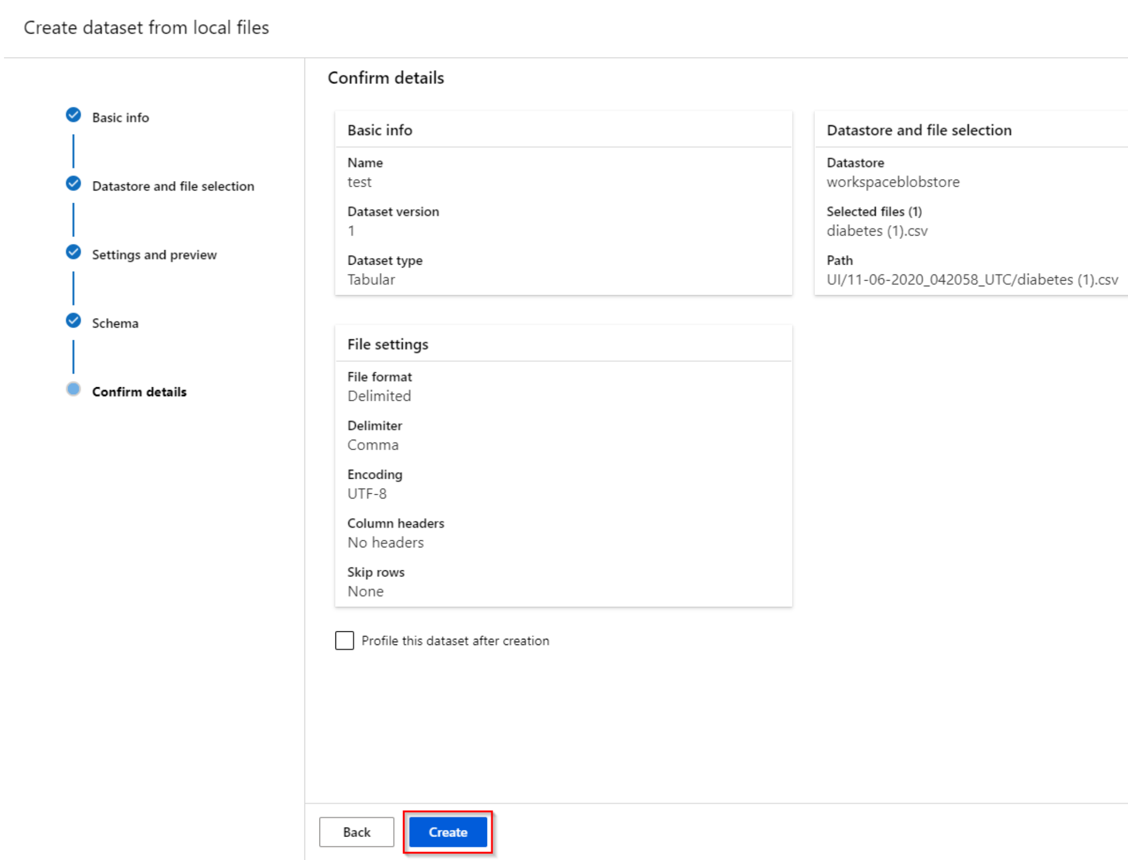




- File format: delimited, plain text, parquet file or JSON Lines

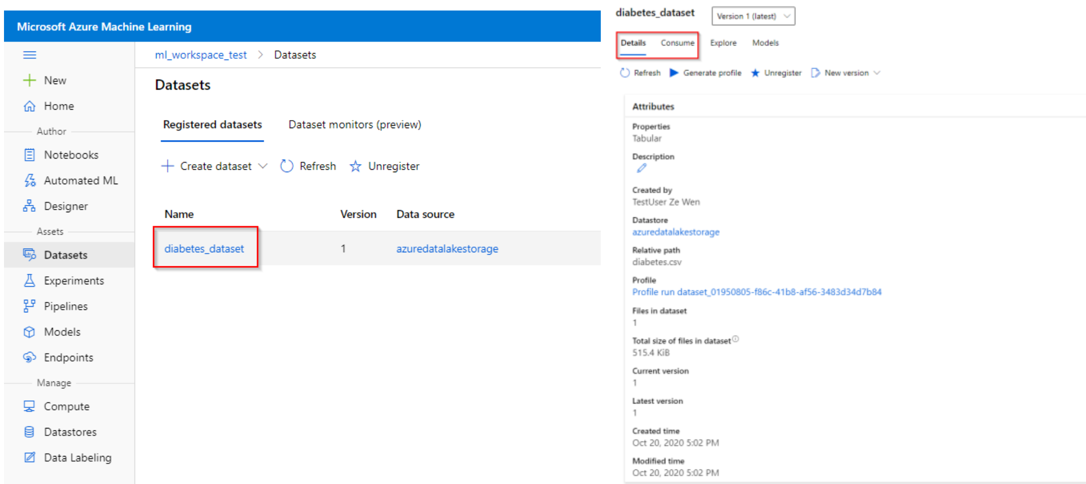



Create dataset from datastore
Basic info
Datastore selection
Settings and preview
Schema
Confirm details
Schema
Include
Column name
Properties
Type
Format settings and example
Path
Not applicable to selected type
String
PatientID
Integer
1354778, 1147438, 1640031
Pregnancies
0, 8, 7
PlasmaGlucose
171, 92, 115
DiastolicBloodPressure
80, 93, 47
TricepsThickness
34, 47, 52
SerumInsulin
23, 36, 35
BMI
Decimal
43.50972593, 21.24057571, 41.5...
DiabetesPedigree
1.213191354, 0.158364981, 0.07...
Age
21, 23, 23
Diabetic
0, 0, 0
Back
Next
Cancel

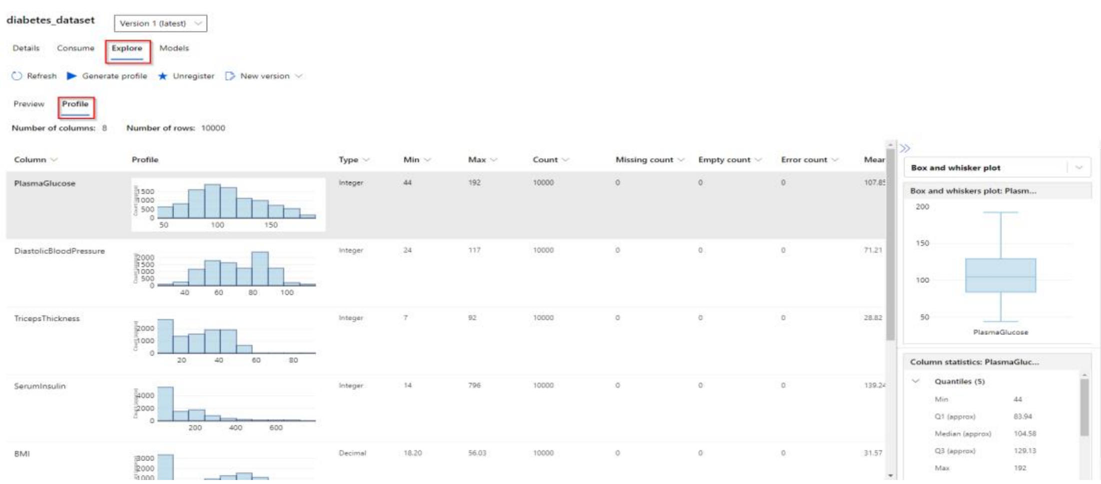

Create dataset from local files
Basic info
Datastore and file selection
Settings and preview
Schema
Confirm details
Confirm details
Basic info
Name
test
Dataset version
1
Dataset type
Tabular
Datastore and file selection
Datastore
workspaceblobstore
Selected files (1)
diabetes (1).csv
Path
UI/11-06-2020_042058_UTC/diabetes (1).csv
File settings
File format
Delimited
Delimiter
Comma
Encoding
UTF-8
Column headers
No headers
Skip rows
None
Profile this dataset after creation
Back
Create

- **Viewing the dataset**

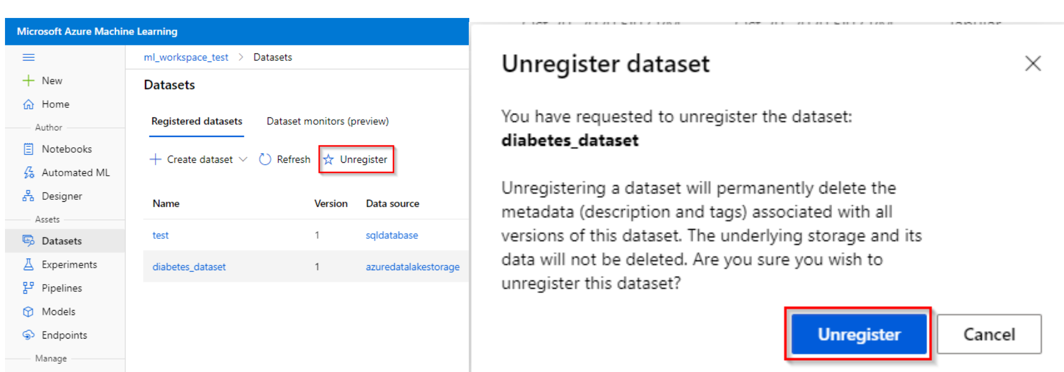
After a dataset is registered, you can obtain information from this dataset in the workspace.



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



- **Unregister a dataset**



3. 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**.

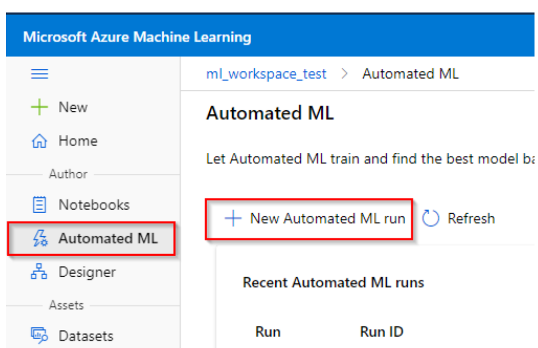
3.1 AutoML

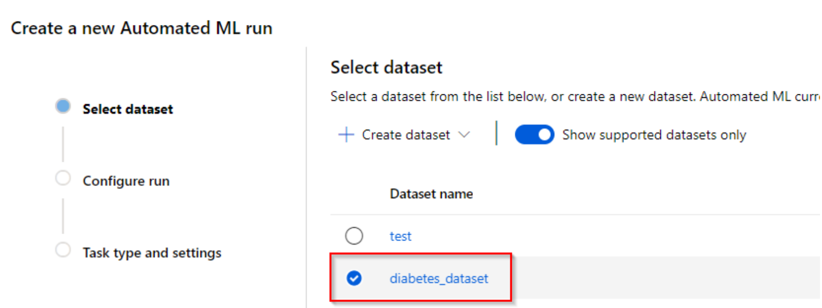
Automated Machine Learning (AutoML) enables you to try multiple algorithms and preprocessing transformations with your data. This, combined with scalable cloud-based compute makes it possible to find the best performing model for your data without the time-consuming manual trial and error that would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

The AutoML feature can handle the following machine learning tasks:

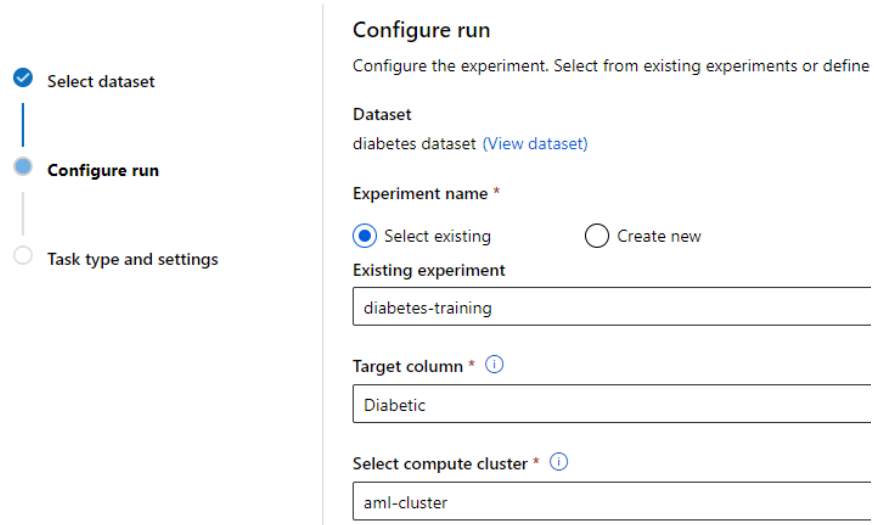
- Classification
- Regression
- Time-series forecasting

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



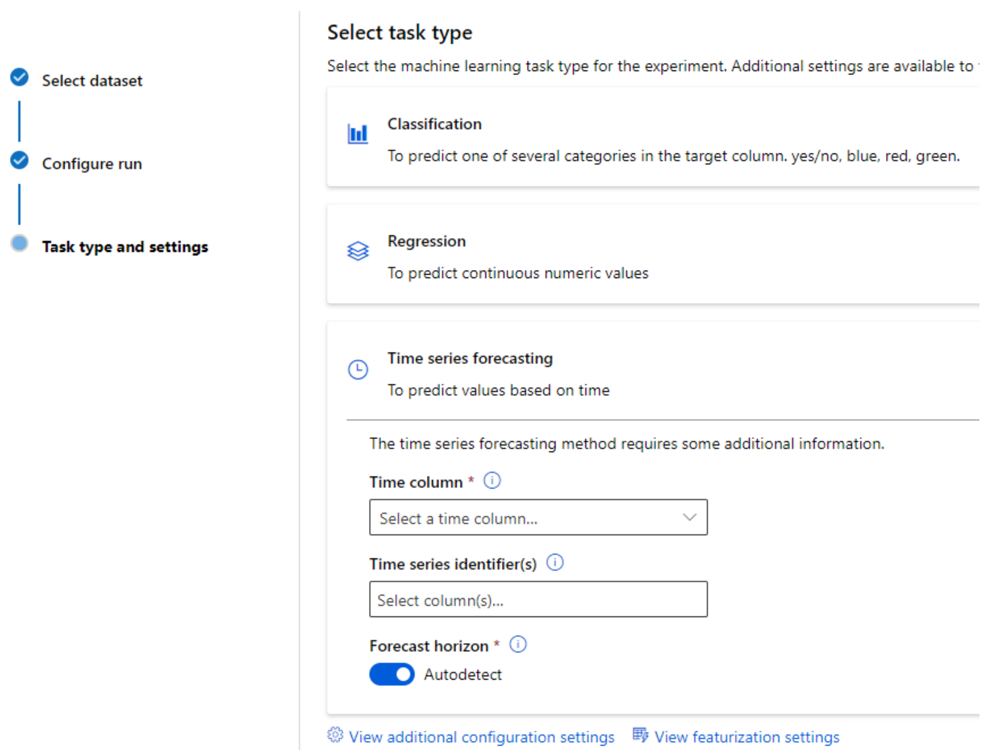


- target column: the column in the dataset that you want the model to predict



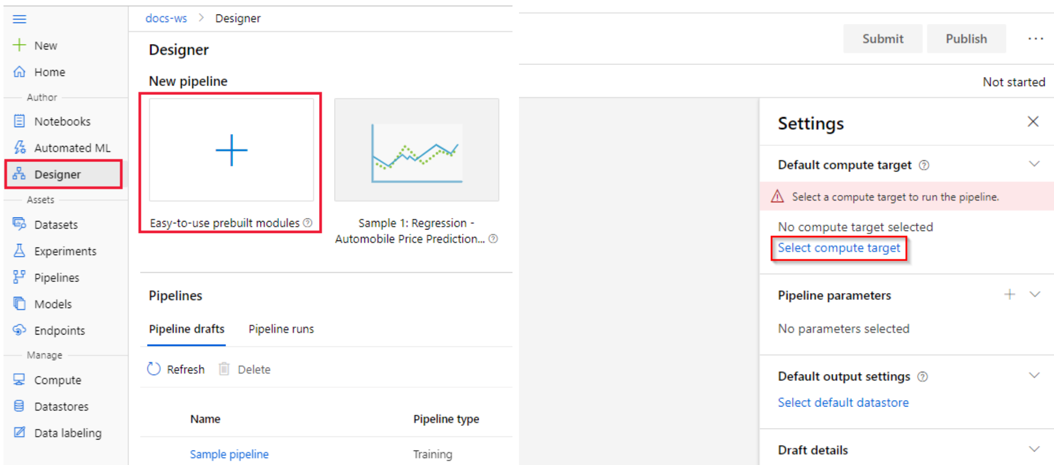
- In the classification category, you can specify deep learning capabilities of AutoML. In the time-series forecasting category, additional information such as time column and forecast horizon need to be provided.

- 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.

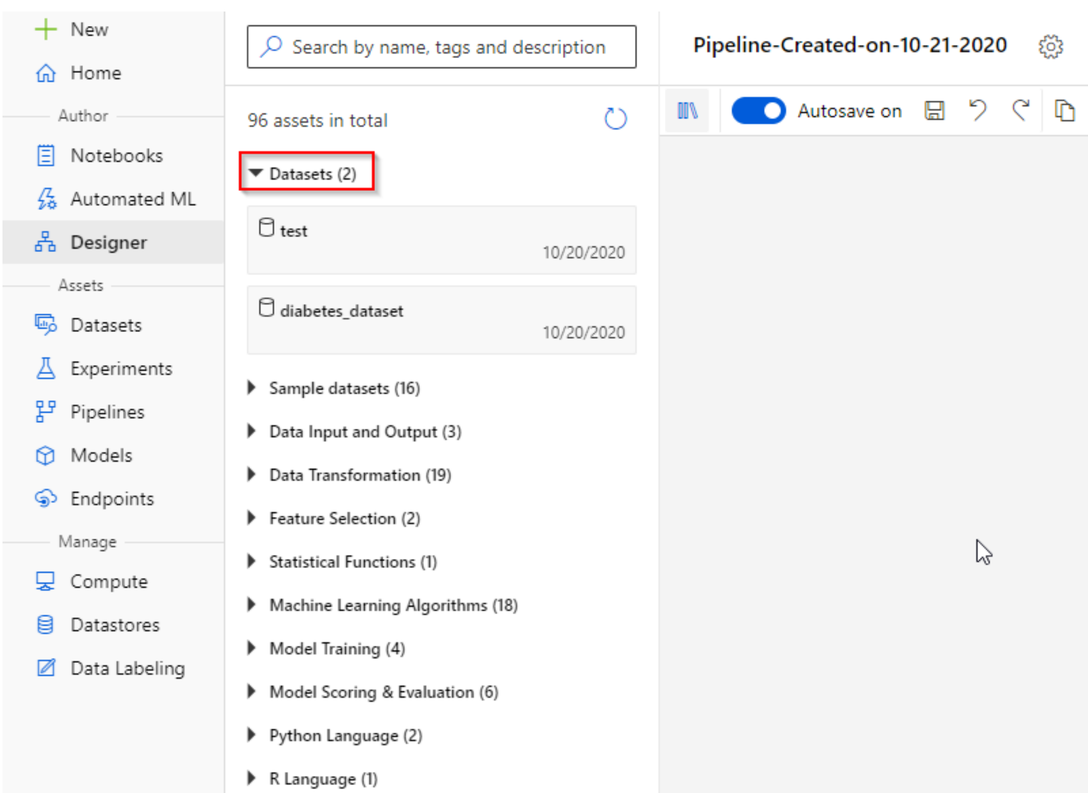


3.2 Designer

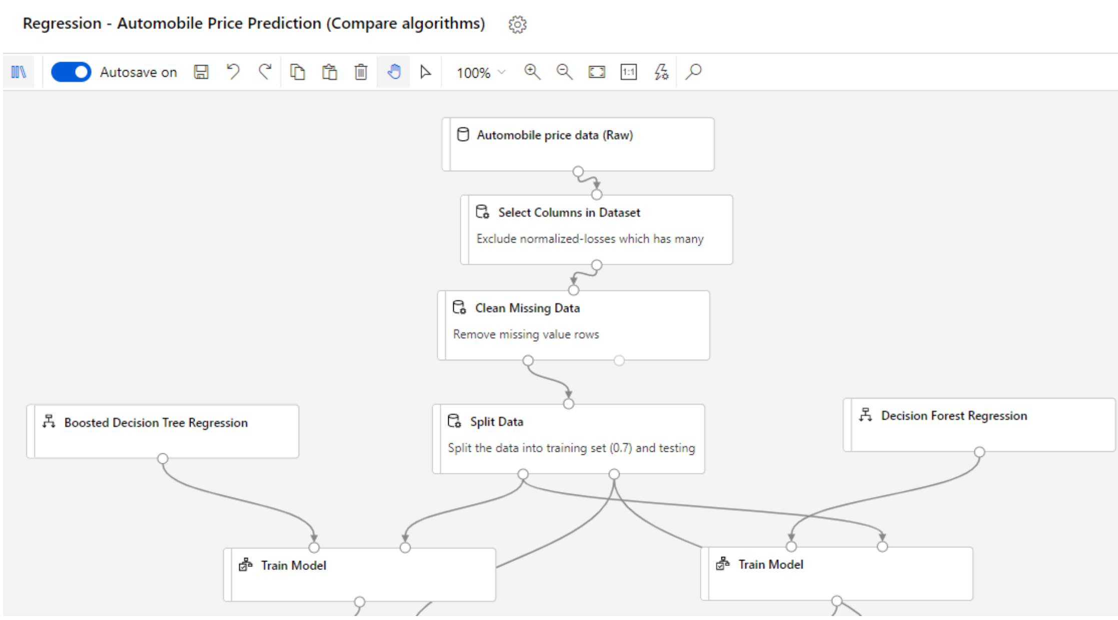
In an enterprise data science process, you'll generally want to separate the overall process into individual tasks, and orchestrate these tasks as pipelines of connected steps. Azure Machine Learning designer lets you visually connect datasets and modules on an interactive canvas to create machine learning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

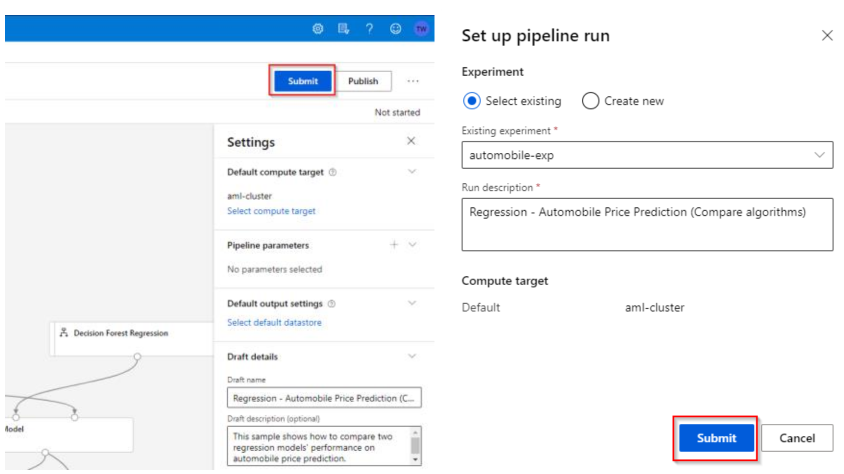


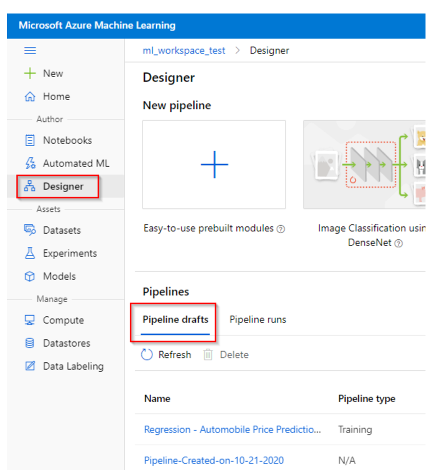
- the different assets/building blocks you can use to pre-process, train and post-process your dataset. In the “Datasets” asset, you will see all the registered datasets in the workspace.



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

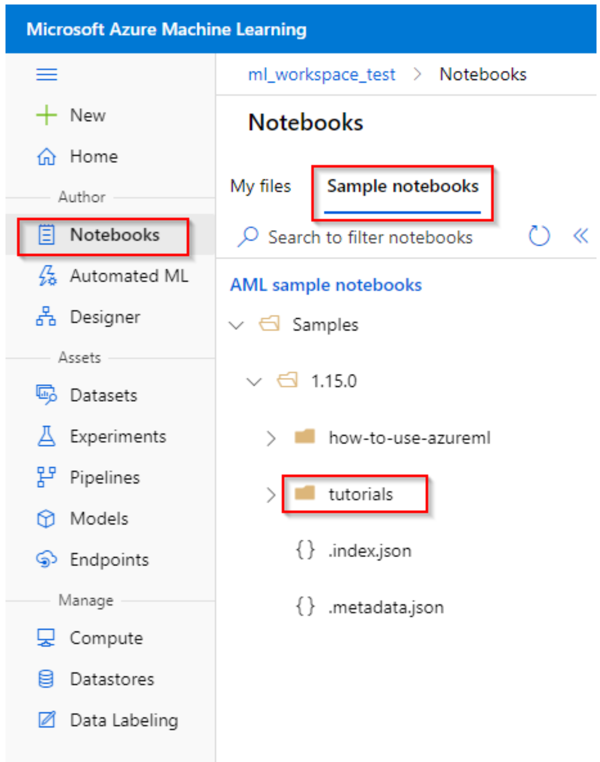






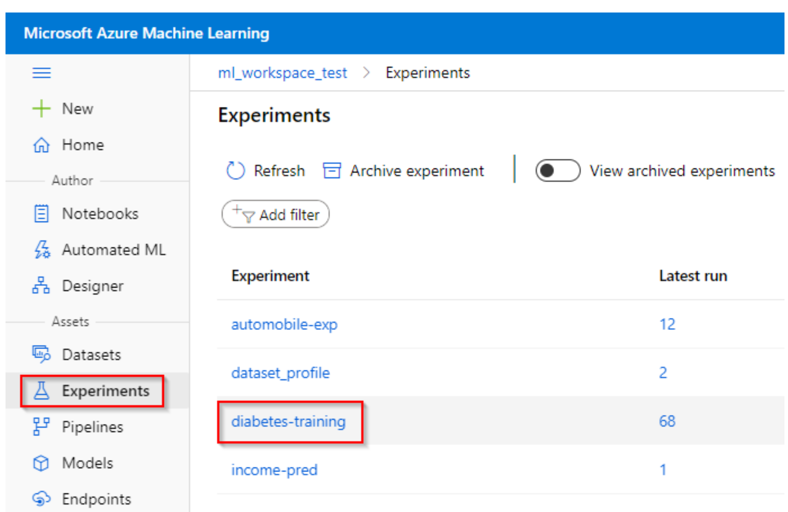
3.3 Notebooks

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.

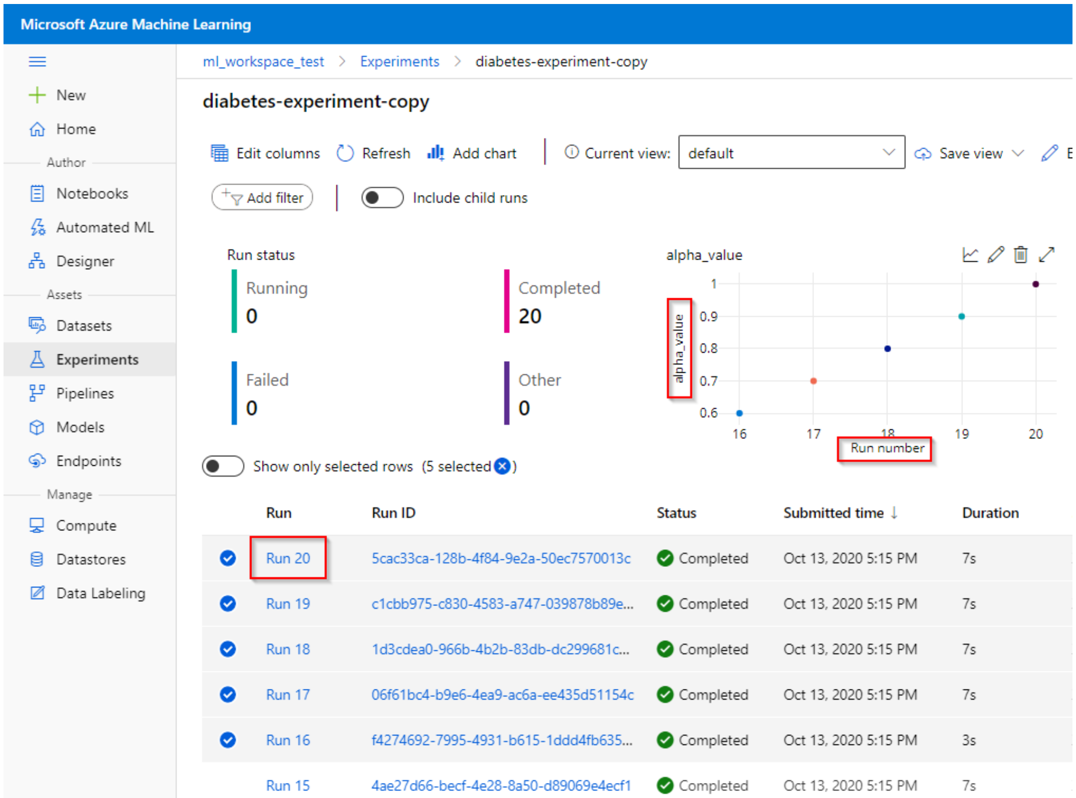


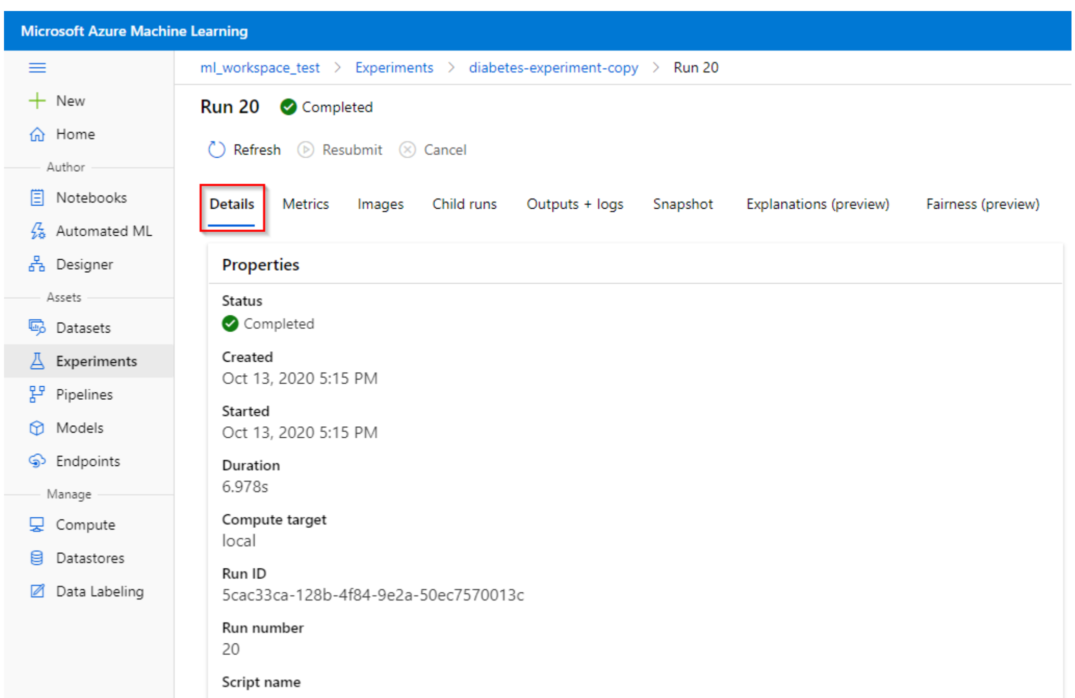
- **Exploring the Experimental runs**

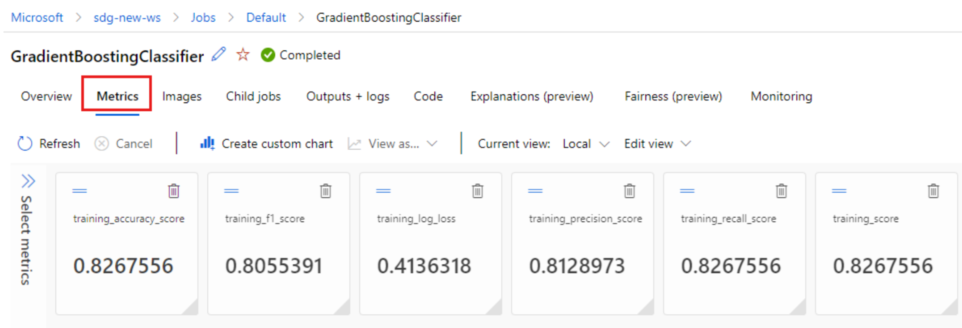
The information on an Azure ML experimental run can be viewed in the “Experiment” tab under the “Assets” category.

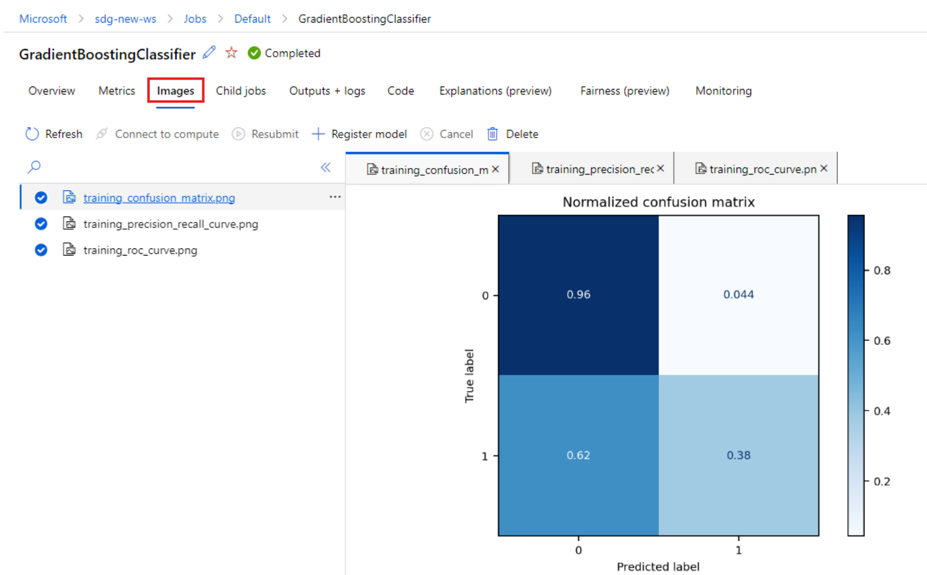


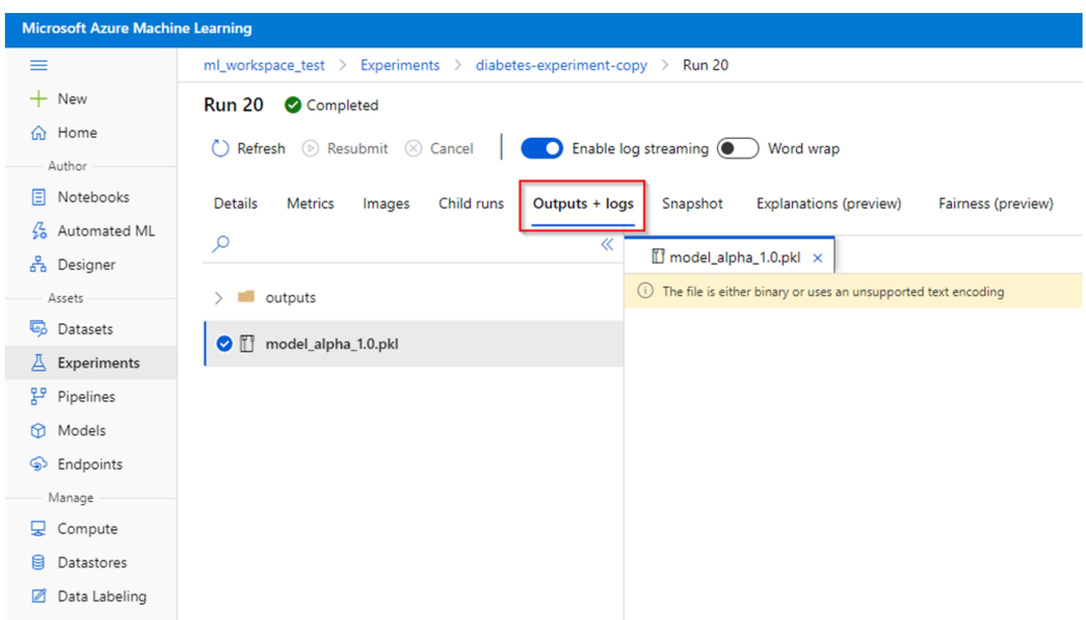
In the Azure ML experiments, you can see which runs are still running, failed or completed. If you assigned any custom, logged metrics to your runs (using the Notebook feature), the evolution of these metrics over different runs can be viewed here.






Microsoft > sdg-new-ws > Jobs > Default > GradientBoostingClassifier
GradientBoostingClassifier
Completed
Overview
Metrics
Images
Child jobs
Outputs + logs
Code
Explanations (preview)
Fairness (preview)
Monitoring
Refresh
Cancel
Create custom chart
View as...
Current view: Local
Edit view
Select metrics
training_accuracy_score
0.8267556
training_f1_score
0.8055391
training_log_loss
0.4136318
training_precision_score
0.8128973
training_recall_score
0.8267556
training_score
0.8267556


Microsoft > sdg-new-ws > Jobs > Default > GradientBoostingClassifier
GradientBoostingClassifier
Completed
Overview
Metrics
Images
Child jobs
Outputs + logs
Code
Explanations (preview)
Fairness (preview)
Monitoring
Refresh
Connect to compute
Resubmit
Register model
Cancel
Delete
training_confusion_matrix.png
training_precision_recall_curve.png
training_roc_curve.png
Normalized confusion matrix
0.96
0.044
0.62
0.38
True label
Predicted label

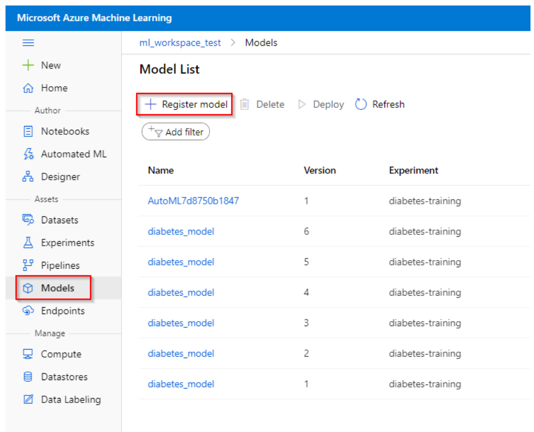

Microsoft Azure Machine Learning
ml_workspace_test > Experiments > diabetes-experiment-copy > Run 20
New
Home
Author
Notebooks
Automated ML
Designer
Assets
Datasets
Experiments
Pipelines
Models
Endpoints
Manage
Compute
Datastores
Data Labeling
Run 20
Completed
Refresh
Resubmit
Cancel
Enable log streaming
Word wrap
Details
Metrics
Images
Child runs
Outputs + logs
Snapshot
Explanations (preview)
Fairness (preview)
outputs
model_alpha_1.0.pkl
The file is either binary or uses an unsupported text encoding

4. Model registration

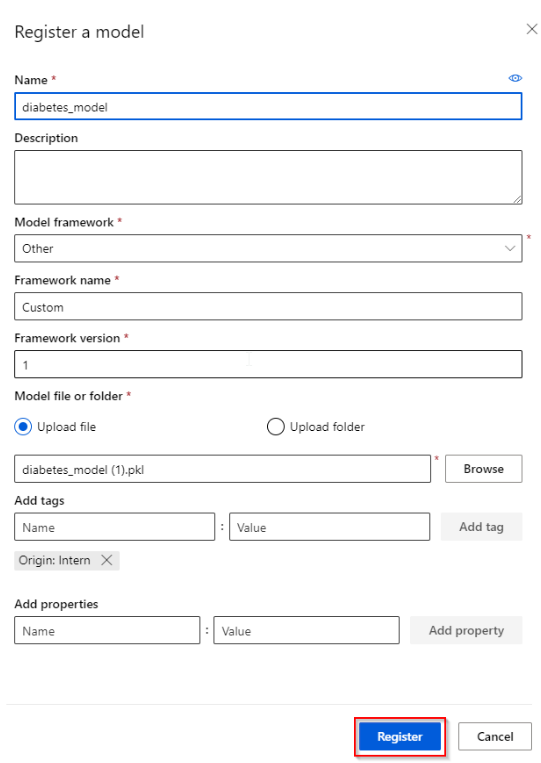
After training the model, you can register it in the workspace from the trained machine learning model runs. Registering a model might be beneficial to manage the machine learning environment of your organization, e.g., be able to reference the model in the Notebook feature.

4.1 Registering a model from registered experimental runs

4.2 Registering a model from a local model file

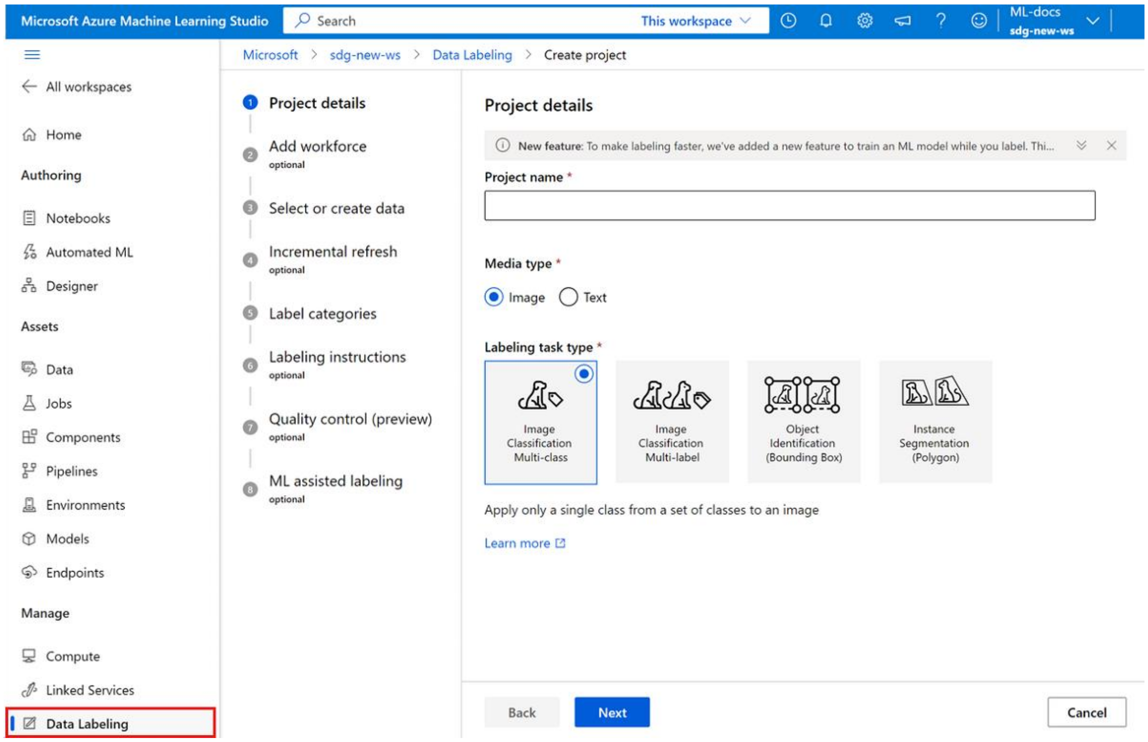


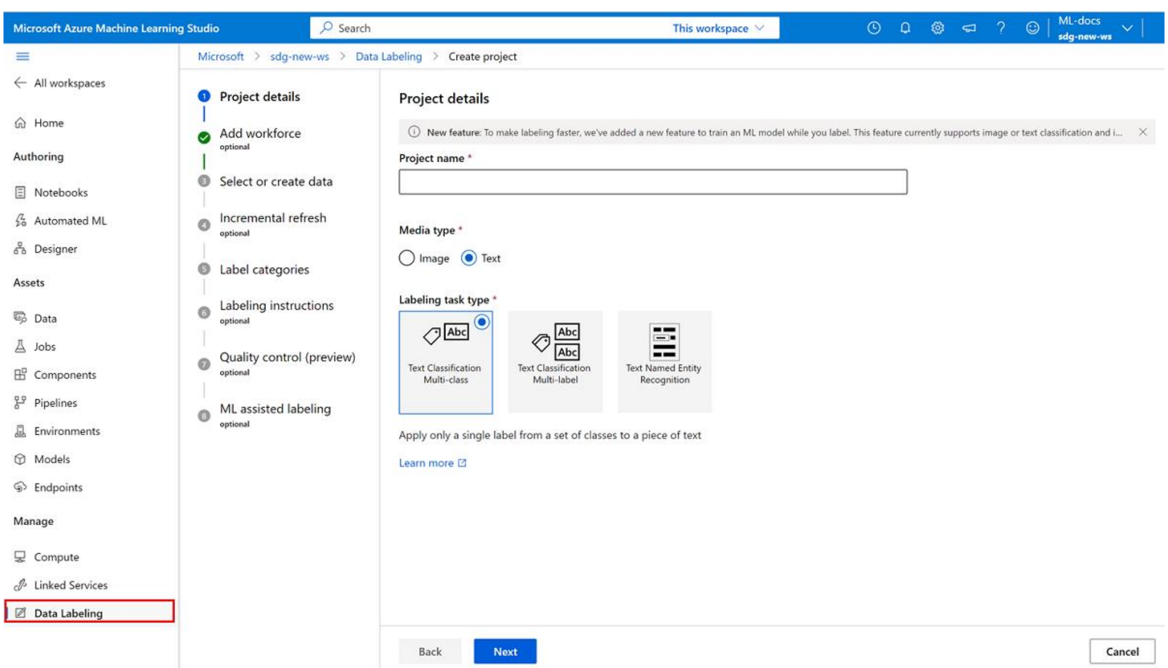
- Giving a name to your model and specify the model framework. Also, provide the framework version (for keeping track of your models) and model file corresponding to the model framework given (e.g. the pkl-file). You can also provide tags and properties that are associated with your model.

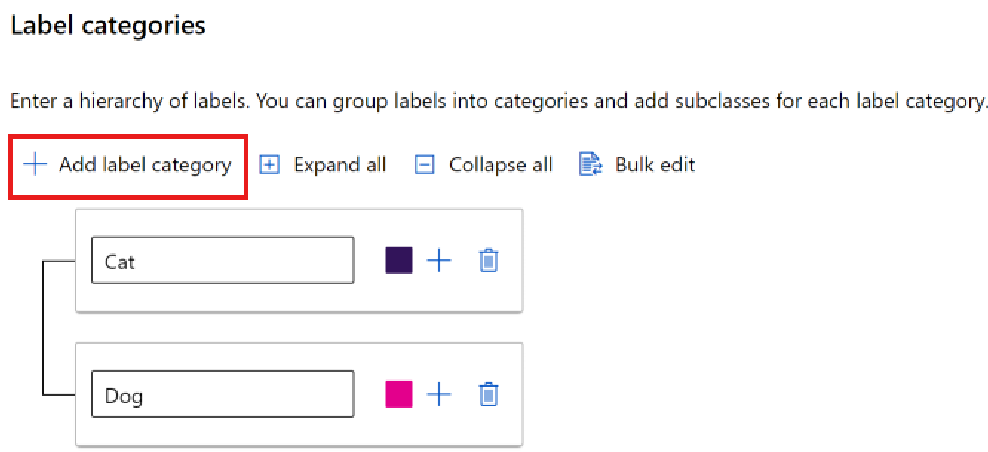


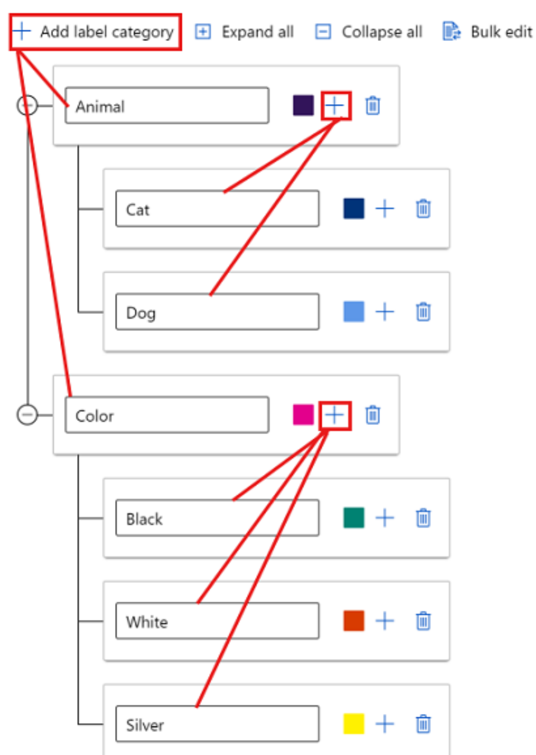
5. Data Labeling

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.







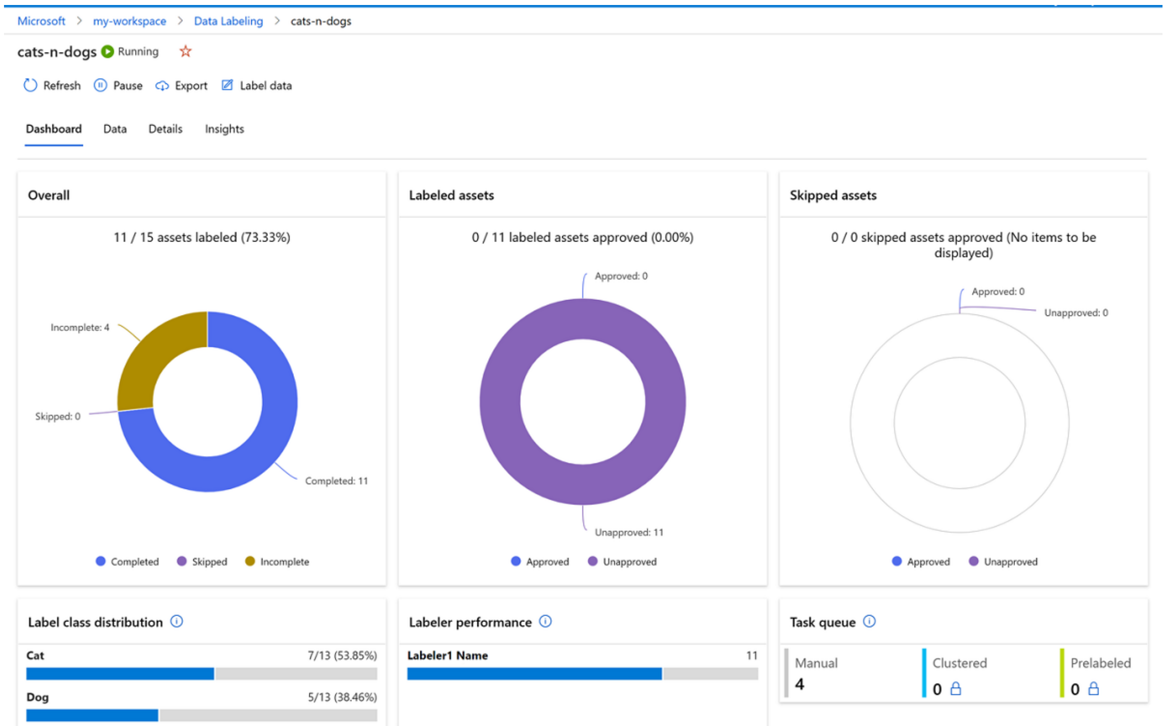


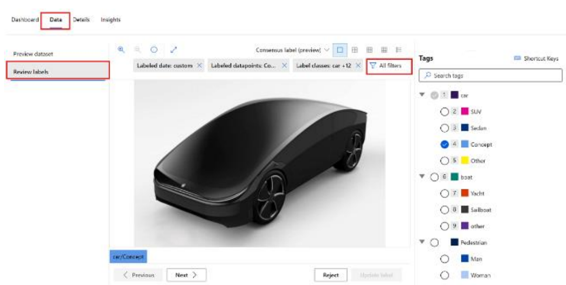
- To accelerate labeling tasks, on the ML assisted labeling page, you can trigger automatic machine learning models. For text, Machine learning (ML)-assisted labeling can handle both file (.txt) and tabular (.csv) text data inputs.

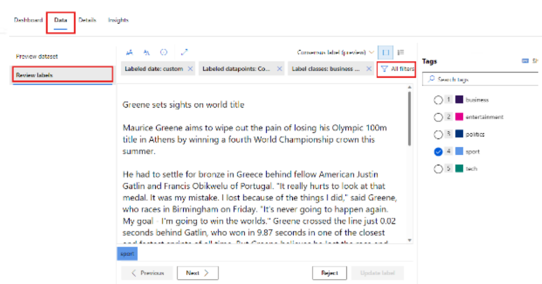
ML-assisted labeling consists of two phases:

- Clustering
- Pre-labeling

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

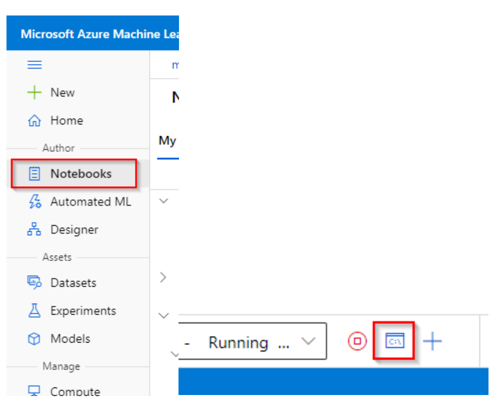






6. Git integration

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.

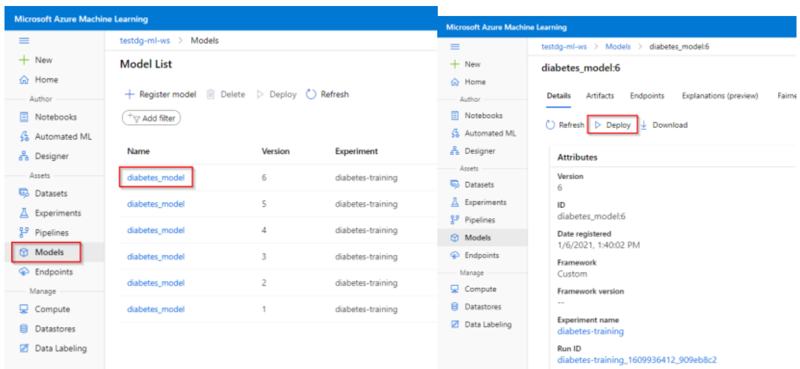


7. Model Deployment

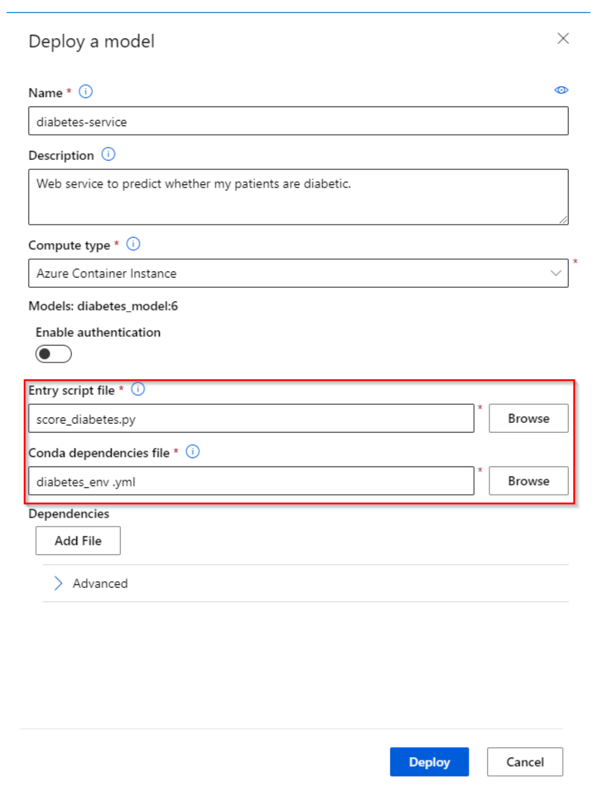
Azure ML Studio has also the feature to deploy a trained machine learning model using a real-time Azure Container Instances (ACIs), a Web Service.

7.1 Deploying a web service

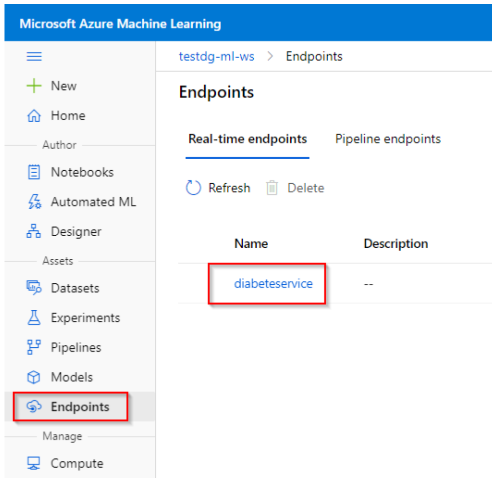
A model can only be deployed after you have registered it in the Azure ML Studio. To deploy a model.

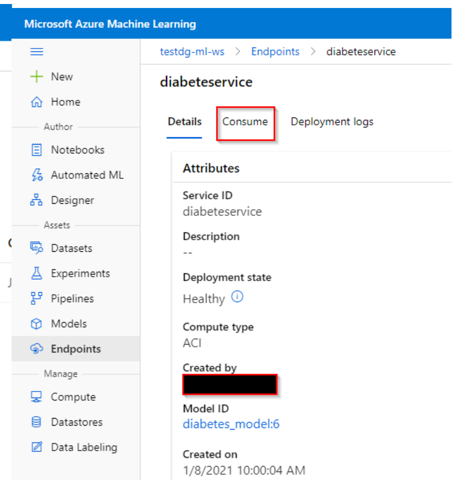


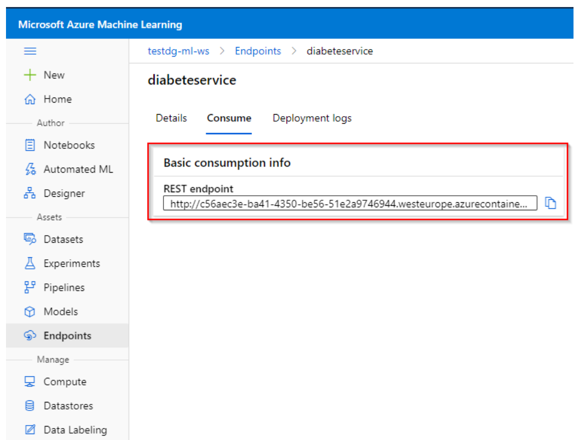
- Provide the entry script and a conda dependencies file. The entry script is used to process the data send to the Web Service. It can be a python file containing an init() and a run(data) function. The conda dependencies files defines the environment used to run the deployed model.



7.2 view a web service







7.3 Deleting a web service