Practice activity: Implementing data storage solutions

Introduction

In this hands-on activity, you will implement a data storage solution for a machine learning project using Azure's storage services. This exercise is designed to give you practical experience in selecting, configuring, and using the most appropriate Azure storage solution based on specific project requirements. By completing this activity, you will better understand how different storage solutions support data management and enhance machine learning workflows.

By the end of this activity, you will be able to:

- Set up and configure a data storage solution in Azure Machine Learning studio.
- Understand best practices for implementing the chosen data storage solution.

Step-by-step guide to implementing data storage solutions

This reading will guide you through the following steps:

- Step 1: Explore data assets.
- Step 2: Understand datastores.
- Step 3: Monitor data with dataset monitors (preview feature).
- Step 4: Import data.
- Step 5: Configure data connections.

Step 1: Explore data assets

Log in to ml.azure.com and navigate to your workspace.

Go to Assets -> Data, where you can view the Data Assets tab.

- Definition: data assets are immutable references to your data created from datastores, local files, public URLs, or open datasets.
- AzureML v2 APIs: data assets created with AzureML v2 APIs cannot be deleted, but they can be up-versioned or archived for better reusability and tracking in machine learning tasks.

- AzureML v1 APIs: data assets created with these APIs are permanently deleted when removed, along with associated metadata.

Step 2: Understand datastores

Datastores securely store connection information for accessing Azure storage services.

- Benefits:
    - Eliminate the need to provide credential information in scripts
    - Simplify the process of connecting to Azure storage
- Default datastores:
    - workspaceworkingdirectory: stores your working directory, including Jupyter notebooks and associated files.
    - workspaceblobstore: used for data related to machine learning jobs, ensuring efficient storage and retrieval.

Use datastores to ensure a secure and seamless integration with Azure Machine Learning tasks.

Step 3: Monitor data with dataset monitors (preview feature)

- Overview: dataset monitors detect data drift between a model's training data and its inference data.
    - Requires enabling model data collection.
    - Note: this feature has been deprecated, and its availability may change in the future.
- Use case: detecting significant changes in the distribution or quality of incoming data that might affect model performance.

Step 4: Import data

- Data import jobs allow you to bring in data from external sources for use in Azure Machine Learning studio.
    - Single job or scheduled: run jobs once or on a defined schedule.
    - Supported sources: import data from Snowflake, AzureSQL, Amazon S3, or other storage solutions.
- Example workflow:
    - Click "Add Import" in the interface.

    - Select the source and provide connection details.
    - Initiate the import to move external data into Azure Machine Learning.
- Imported data can then be used directly in training tasks or preprocessing steps, streamlining workflows.

Step 5: Configure data connections

- Definition: data connections enable importing data into Azure Machine Learning by integrating external repositories or services.
- Supported integrations:
    - Git repositories such as GitHub for version-controlled data and scripts
    - Container registries or Python package feeds for managing dependencies and libraries
- Workflow:
    - Set up data connections through the Azure Machine Learning studio interface or via API calls.
    - Leverage these connections to centralize data management for your projects.

Conclusion

In this activity, we explored the following key data storage solutions in Azure Machine Learning studio:

- Data assets: immutable references for structured data management
- Datastores: secure connections to Azure storage services for seamless integration
- Dataset monitors: tools for tracking data drift and ensuring model reliability
- Data importing: flexible options for incorporating external datasets
- Data connections: integrations for external repositories and services to centralize data workflows.

By selecting the right storage solution and following best practices, you can improve the efficiency, scalability, and security of your machine learning projects.