Learn how to create and use components to build pipeline in Azure Machine Learning. Run and schedule Azure Machine Learning pipelines to automate machine learning workflows.

**Learning objectives**

In this module, you'll learn how to:

* Create components.
* Build an Azure Machine Learning pipeline.
* Run an Azure Machine Learning pipeline.

-- **Introducción a los Pipelines en Azure Machine Learning**

En Azure Machine Learning, puedes experimentar en notebooks y entrenar (y volver a entrenar) modelos de machine learning ejecutando scripts como trabajos. En un proceso de ciencia de datos empresarial, es esencial dividir el proceso general en tareas individuales y agruparlas en **pipelines**. Los pipelines son fundamentales para implementar una solución efectiva de **Machine Learning Operations (MLOps)** en Azure.

**¿Qué son los Pipelines?**

Los **pipelines** en Azure Machine Learning permiten organizar y automatizar tareas relacionadas con el entrenamiento de modelos de machine learning. Al separar el proceso en componentes de tareas individuales, facilitas la reutilización y el intercambio de código. Esto no solo mejora la eficiencia, sino que también ayuda a mantener un enfoque organizado en el desarrollo de modelos.

**Componentes de un Pipeline**

Un pipeline en Azure Machine Learning puede incluir diversos componentes, tales como:

* **Datos de entrada**: Fuentes de datos que alimentan el modelo.
* **Transformaciones**: Procesos de preparación de datos, como limpieza y normalización.
* **Entrenamiento de modelos**: Etapas donde se entrenan y ajustan los modelos de machine learning.
* **Evaluación**: Fases en las que se evalúa el rendimiento del modelo.
* **Despliegue**: Implementación del modelo en un entorno de producción.

**Ejecución de Pipelines**

Los pipelines se ejecutan como **pipeline jobs**. Al definir un pipeline, puedes:

1. **Crear componentes**: Define cada tarea o paso como un componente reutilizable.
2. **Combinar componentes**: Agrupa los componentes para formar un pipeline completo.
3. **Ejecutar el pipeline**: Inicia el pipeline como un trabajo para automatizar el flujo de trabajo de machine learning.

**Interacción entre Pipelines**

Es importante tener en cuenta que el término "pipeline" se utiliza de manera extensiva en varios dominios, incluidos el machine learning y la ingeniería de software. En Azure DevOps o GitHub, un pipeline puede referirse a tareas de construcción y liberación, mientras que en Azure Synapse Analytics, un pipeline se utiliza para definir procesos de ingestión y transformación de datos.

Aunque este módulo se centra en los pipelines de Azure Machine Learning, es posible que los pipelines de diferentes servicios interactúen entre sí. Por ejemplo, un pipeline de Azure DevOps o Azure Synapse Analytics puede desencadenar un pipeline de Azure Machine Learning, lo que permite una integración más fluida en todo el ciclo de vida del desarrollo de software y modelos de machine learning.

-- **Creación de Componentes en Azure Machine Learning**

Los componentes en Azure Machine Learning te permiten crear scripts reutilizables que pueden ser compartidos fácilmente entre usuarios dentro del mismo espacio de trabajo. Esto es especialmente útil al construir pipelines de machine learning y al preparar tu código para la producción.

**¿Por qué usar componentes?**

1. **Construcción de pipelines**: Facilitan la creación de flujos de trabajo automatizados.
2. **Compartición de código**: Permiten a otros usuarios reutilizar código ya preparado y probado.

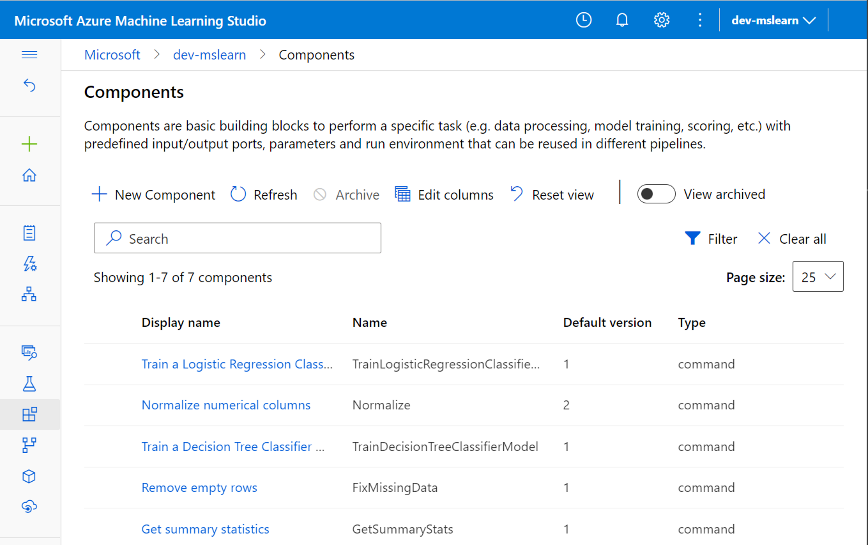
**Estructura de un Componente**

Un componente se compone de tres partes principales:

1. **Metadatos**: Incluye el nombre del componente, la versión, etc.
2. **Interfaz**: Especifica los parámetros de entrada esperados (como un conjunto de datos o hiperparámetros) y los resultados esperados (como métricas y artefactos).
3. **Comando, código y entorno**: Define cómo ejecutar el código.

**Creación de un Componente**

Para crear un componente, necesitas dos archivos:

1. Un script que contenga el flujo de trabajo que deseas ejecutar.
2. Un archivo YAML que defina los metadatos, la interfaz y el comando, código y entorno del componente.

En este ejemplo, crearemos un componente que prepara datos mediante un script de Python (prep.py):

**Script prep.py**

Texto

Descripción generada automáticamente

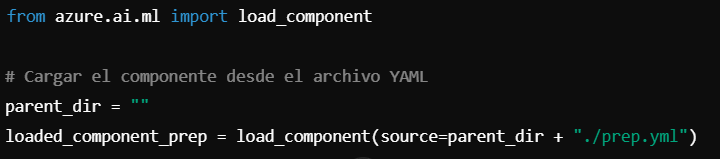
**Archivo YAML prep.yml**

Texto

Descripción generada automáticamente

**Cargar y Usar el Componente**

Una vez que tengas el script y el archivo YAML, puedes cargar el componente usando el siguiente código:



**Registrar un Componente**

Para hacer que los componentes sean accesibles para otros usuarios en el espacio de trabajo, debes registrarlos en Azure Machine Learning. Puedes registrarlos con el siguiente código:

prep = ml\_client.components.create\_or\_update(loaded\_component\_prep)

-- **Creación de un Pipeline en Azure Machine Learning**

En Azure Machine Learning, un **pipeline** es un flujo de trabajo que organiza tareas de machine learning, donde cada tarea se define como un componente. Los componentes pueden ser ejecutados en secuencia o en paralelo, lo que permite construir lógicas de flujo sofisticadas para orquestar operaciones de machine learning.

**Estructura de un Pipeline**

1. **Definición**: Un pipeline se define en un archivo YAML que incluye el nombre del trabajo del pipeline, entradas, salidas y configuraciones.
2. **Ejecución**: Un pipeline puede ejecutarse como un trabajo, donde cada componente se ejecuta como un subtrabajo dentro del trabajo del pipeline.

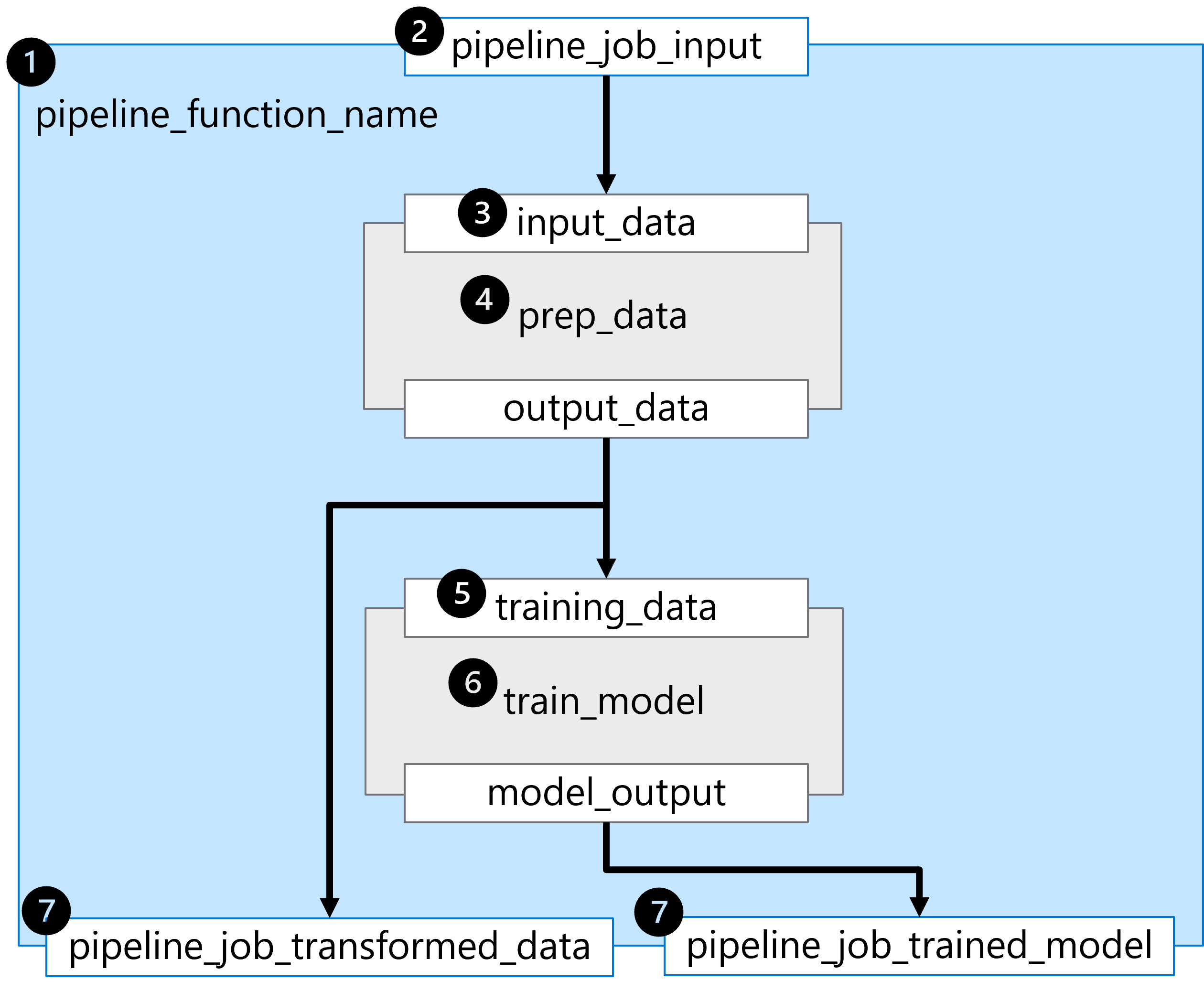
**Cómo Construir un Pipeline**

Para construir un pipeline que prepare datos y luego entrene un modelo, puedes utilizar el siguiente código:

Texto

Descripción generada automáticamente

**Pasos Explicados**

1. **Definición de la función**: Se define la función pipeline\_function\_name, que crea el pipeline.
2. **Parámetros de entrada**: La función espera un parámetro de entrada pipeline\_job\_input que se utilizará como entrada para los pasos del pipeline.
3. **Componente de preparación**: El primer paso del pipeline utiliza el componente loaded\_component\_prep y pasa los datos de entrada.
4. **Componente de entrenamiento**: El segundo paso utiliza el componente loaded\_component\_train y recibe los datos transformados del primer paso como entrada.
5. **Salidas del pipeline**: Se devuelven dos salidas: pipeline\_job\_transformed\_data y pipeline\_job\_trained\_model.

**Llamar al Pipeline**

Texto

Descripción generada automáticamentePara ejecutar el pipeline con un activo de datos registrado como entrada, puedes hacer lo siguiente:

**Revisar la Configuración del Pipeline**

Al ejecutar la función @pipeline(), se genera un archivo YAML que representa la configuración del pipeline. Puedes revisar este archivo imprimiendo el objeto pipeline\_job:

print(pipeline\_job)

**Ejemplo de Salida YAML**

Texto

Descripción generada automáticamenteLa salida se verá similar a esto:

--- **Ejecución de un Pipeline Job en Azure Machine Learning**

Después de construir un pipeline basado en componentes, puedes ejecutar el flujo de trabajo como un pipeline job. A continuación se explica cómo configurar y ejecutar un pipeline, y cómo programar su ejecución automática.

**Configuración de un Pipeline Job**

Antes de ejecutar el pipeline, puedes ajustar configuraciones específicas en el objeto pipeline\_job:

1. **Cambio del modo de salida**: Define cómo se almacenarán los datos de salida del pipeline.

Texto

Descripción generada automáticamente

1. **Configuración de cómputo predeterminado**: Define un recurso de cómputo predeterminado para los componentes en caso de que no tengan uno especificado.

Texto

Descripción generada automáticamente

1. **Almacenamiento predeterminado**: Define el almacén de datos en el que se almacenarán todas las salidas.



Para revisar la configuración del pipeline, imprime el objeto pipeline\_job:

print(pipeline\_job)

**Ejecución del Pipeline Job**

Cuando hayas terminado de configurar el pipeline, puedes ejecutarlo como un pipeline job:

Texto

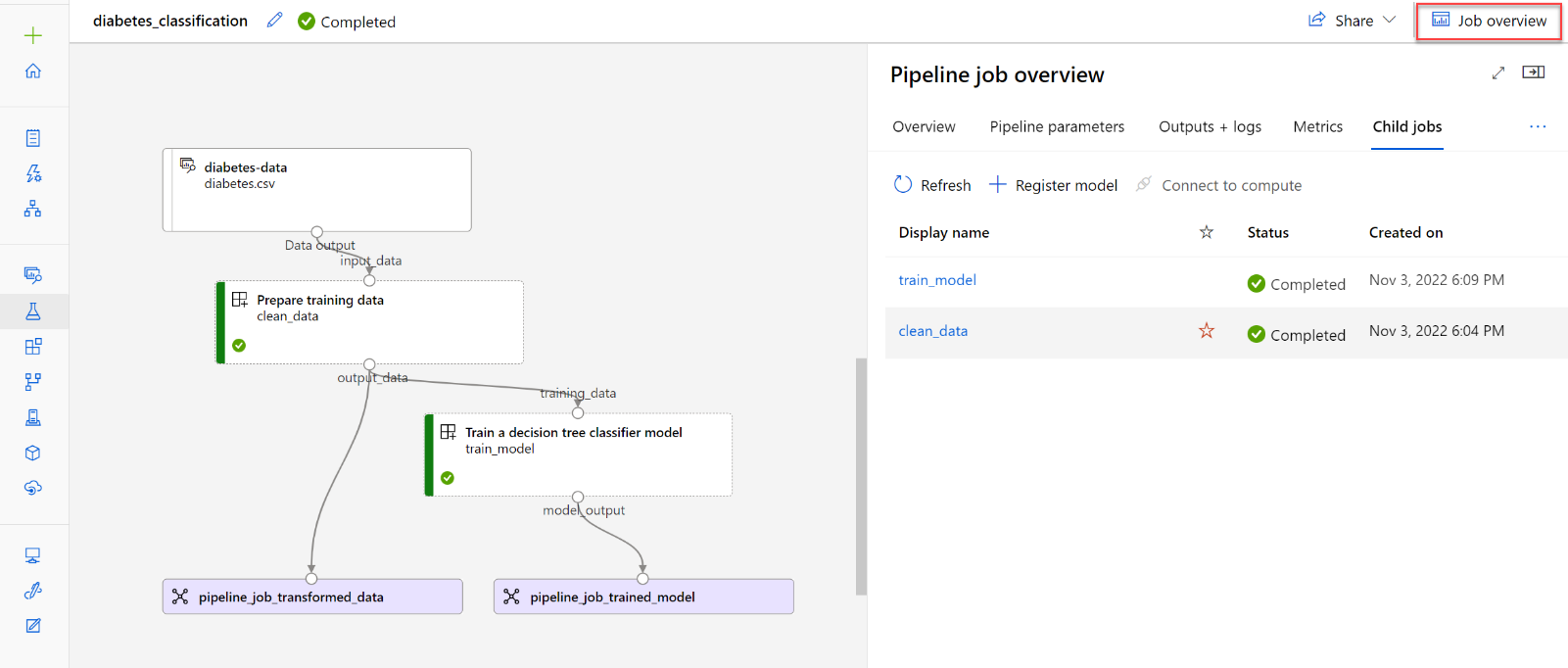
Descripción generada automáticamente

Una vez que envíes el pipeline job, Azure Machine Learning creará un nuevo trabajo en el workspace, incluyendo subtrabajos que representan la ejecución de cada componente. En el Azure Machine Learning Studio, puedes ver una representación gráfica del pipeline y explorar los parámetros, salidas y subtrabajos.

**Solución de Problemas**

Para depurar un pipeline fallido:

* Revisa las **salidas y registros** del pipeline job y sus subtrabajos.
* Si el problema es de configuración del pipeline, consulta los registros del pipeline job.
* Si el problema es con un componente específico, revisa los registros del subtrabajo correspondiente.



**Programación de un Pipeline Job**

Un pipeline es ideal para producción, especialmente cuando necesitas automatizar el reentrenamiento de un modelo. Puedes crear una programación para ejecutar el pipeline en intervalos regulares.

**Crear una Programación Basada en Tiempo**

Texto

Descripción generada automáticamentePara crear una programación que ejecute el pipeline cada minuto, usa RecurrenceTrigger:

Texto

Descripción generada automáticamenteLuego, asocia el pipeline job con la programación:

Los trabajos ejecutados por la programación tendrán nombres prefijados con el nombre de la programación. Puedes revisar estos trabajos en el Azure Machine Learning Studio.

**Deshabilitar y Eliminar una Programación**

Para detener y eliminar una programación:

Texto

Descripción generada automáticamente

