



códigofacilito

Módulo 4 – Optimización del entrenamiento de modelos con Azure Machine Learning

Ing Cinthya Cabanzo





códigofacilito

Realización del ajuste de hiperparámetros con Azure Machine Learning

Ing. Cinthya Cabanzo





>_ Agenda

- > Introducción
- > Definición del espacio de búsqueda
- > Configuración de un método de muestreo
- > Configuración de la terminación anticipada
- > Uso de un trabajo de barrido para el ajuste de hiperparámetros
- > Ejercicio: Ejecución de un trabajo de barrido





>_ Introducción

Los **hiperparámetros** son parámetros ajustables que permiten controlar el proceso de entrenamiento de un modelo

El **ajuste de hiperparámetros**, también denominado **optimización de hiperparámetros** es el proceso de encontrar la configuración de hiperparámetros que produzca el mejor rendimiento.

En **Azure Machine Learning**, puede ajustar los hiperparámetros enviando un script como un **trabajo de barrido**. Un trabajo de barrido ejecutará una prueba para cada combinación de hiperparámetros que se va a probar.





> Definición del espacio de búsqueda

Nos permite que los **hiperparámetros** estén dentro de un rango específico.

Los hiperparámetros pueden ser discretos o continuos



➊ Ejemplo de Espacio de Búsqueda

```
from azure.ai.ml.sweep import Choice, Normal  
  
command_job_for_sweep = command_job(  
    talla_pantalon=Choice(values=[16, 32, 34]),  
    nota_examen=Normal(mu=10, sigma=3),  
)  
    ↑  
Hiperparámetro  
    ↑  
Espacio de Búsqueda
```



> Hiperparámetros Discretos

command_job()

Se especifican con un objeto `Choice` entre valores discretos (rangos, listas) o distribuciones discretas.

```
number_of_hidden_layers=Choice(values=range(1,5))
```

```
batch_size=Choice(values=[16, 32, 64, 128])
```

```
learning_rate=QNormal(mu=10, sigma=3,q=1)
```

```
keep_probability=QUniform(min_value=0.05, max_value=0.1,q=1)
```





> Hiperparámetros Continuos

command_job()

Se especifican distribuciones continuas, dado que tienen un número infinito de posibilidades.

```
temperature=Normal(mu=20, sigma=3)
```

```
velocity=Uniform(min_value=5, max_value=100)
```



● Ejemplo de Espacio de Búsqueda

```
from azure.ai.ml.sweep import Choice, Normal
```

```
command_job_for_sweep = command_job(  
    talla_pantalon=Choice(values=[16, 32, 34]), ← Discreto  
    nota_examen=Normal(mu=10, sigma=3), ← Continuo  
)
```

↑
Hiperparámetro

↑
Espacio de Búsqueda





Ejemplo de Esquema YAML del trabajo de Barrido para definir el espacio de Búsqueda

```
search_space:  
  talla_pantalon:  
    type: choice  
    values: [16,32,34]
```

```
  nota_examen:  
    type: normal  
    mu: 10  
    sigma: 3
```

Hiperparámetro

Espacio de Búsqueda



Importante:

En YAML los espacios son importantes porque dan jerarquía



> Configuración de un método de muestreo

sweep()

Cuadricula -> 100% combinaciones -> solo discretos

sampling_algorithm = "grid"

Aleatorio -> Aleatorio -> discretos & continuos

sampling_algorithm = "random"

Aleatorio Sobol - Aleatorio resultados reproducibles -> discretos & continuos

sampling_algorithm = RandomSamplingAlgorithm(seed=123, rule="sobol")

Bayesiano - Algoritmo de optimización bayesiano (histórico)->discretos & continuos (choice, uniform y quniform)

sampling_algorithm = "bayesian"





> Configuración del objetivo del barrido

```
sweep()
```

primary_metric -> metrica principal

```
primary_metric="accuracy"
```

goal-> objetivo -> maximizar o minimizar

```
goal="Maximize"
```

```
sweep_job =  
command_job_for_sweep.sweep(  
    compute="cpu-cluster",  
    sampling_algorithm = "bayesian",  
    primary_metric="accuracy",  
    goal="Maximize",  
)
```

Registrar métrica de ajuste

```
import mlflow  
mlflow.log_metric("accuracy",  
float(val_accuracy))  
.....
```



> Configuración de la terminación anticipada

Sirve para **detener** un trabajo de barrido en función del rendimiento de los modelos.

Usada cuando el número de **pruebas** es **infinito**, ej: al trabajar con hiperparámetros continuos y un método de muestreo aleatorio o bayesiano





> Configuración de la terminación anticipada

evaluation_interval-> frecuencia con que se aplica la directiva

delay_evaluation-> retrasa la primera evaluación de la directiva un número especificado de intervalos.

```
sweep_job.early_termination = BanditPolicy(  
    slack_factor = 0.1,  
    delay_evaluation = 5,  
    evaluation_interval = 1)
```





>_ Tipos de directivas de terminación anticipada

- Bandidos
- Mediana de detención
- Selección de truncamiento
- Sin directiva de terminación





> Directiva de bandidos

BanditPolicy

Basada en el [factor de demora \(slack_factor\)](#) o la [cantidad de demora \(slack_amount\)](#) y el intervalo de frecuencia y retraso para la evaluación.

La directiva de bandidos finaliza un trabajo cuando la métrica principal no se encuentra dentro del factor de demora o la cantidad de demora especificados del mejor trabajo.

```
sweep_job.early_termination =  
BanditPolicy(slack_factor = 0.1, delay_evaluation  
= 5, evaluation_interval = 1)
```





> Directiva de bandidos

BanditPolicy

slack_amount=

|_| Distancia **absoluta** permitida desde la ejecución con mejor rendimiento. El valor predeterminado es 0.

slack_factor=Proporción de la distancia

% permitida desde la ejecución con mejor rendimiento. El valor predeterminado es 0.





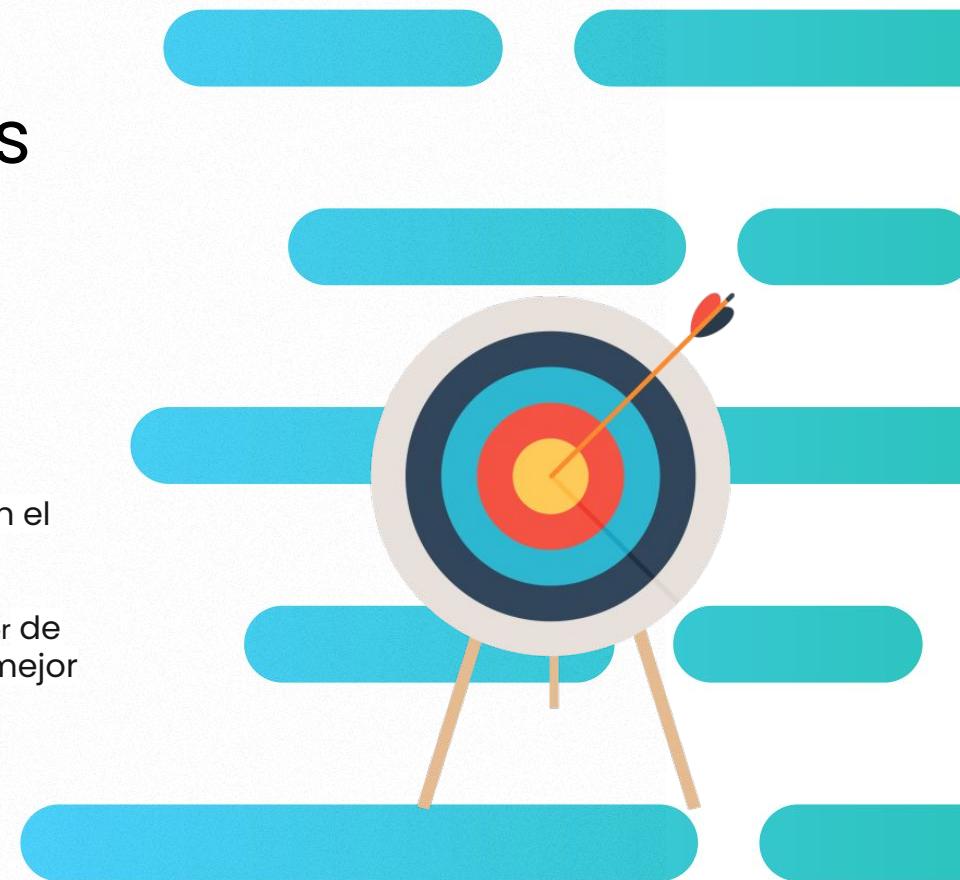
> Directiva de bandidos

BanditPolicy

Por ejemplo, imagine que se aplica una directiva de bandidos en el intervalo 10.

Suponga que el trabajo con el mejor rendimiento en el intervalo 10 informa de una métrica principal de 0,8 con el objetivo de maximizarla.

Si la directiva se especifica con un elemento slack_factor de 0,2, se finalizarán los trabajos de entrenamiento cuya mejor métrica en el intervalo 10 sea **inferior a $0,66$ ($0,8/(1 + slack_factor)$)**.





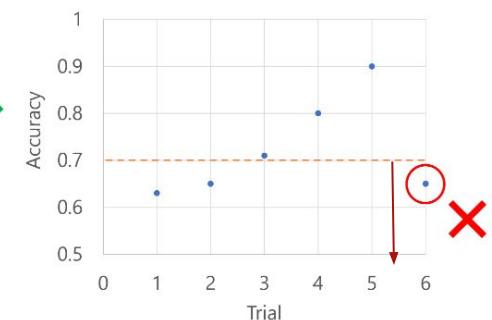
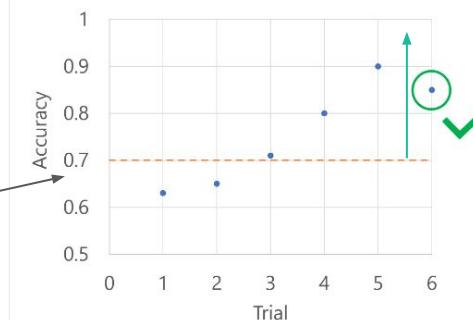
> Directiva de bandidos

BanditPolicy

Imagine que la métrica principal es la precisión del modelo.

Cuando, después de las cinco primeras pruebas, el modelo que mejor funciona tiene una precisión de 0,9, cualquier modelo nuevo tiene que funcionar mejor que $(0,9 - 0,2)$ o 0,7.

Si la precisión del nuevo modelo es superior a 0,7, el trabajo de barrido continuará. Si el nuevo modelo tiene una puntuación de precisión inferior a 0,7, la directiva finalizará el trabajo de barrido.





> Directiva de mediana de detención

MedianStoppingPolicy

Basada en las **medias móviles** de las métricas principales notificadas por los trabajos.

Esta directiva calcula las medias móviles en todos los trabajos de entrenamiento y detiene los trabajos cuyo valor de métrica principal sea **peor que la mediana de los promedios**.





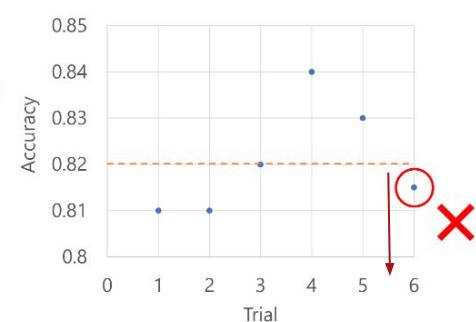
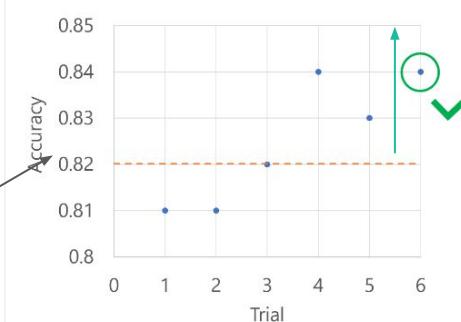
> Directiva de mediana de detención

MedianStoppingPolicy

Imagine que la métrica principal es la precisión del modelo.

Cuando se registra la precisión para la sexta prueba, la métrica debe ser mayor que la mediana de las puntuaciones de precisión hasta el momento.

Supongamos que la mediana de las puntuaciones de precisión hasta ahora es 0,82. Si la precisión del nuevo modelo es superior a 0,82, el trabajo de barrido continuará. Si el nuevo modelo tiene una puntuación de precisión inferior a 0,82, la directiva detendrá el trabajo de barrido y no se entrenará ningún modelo nuevo.





> Directiva de selección de truncamiento

TruncationSelectionPolicy

Una directiva de selección de truncamiento cancela el X % de ejecuciones con menor rendimiento en cada intervalo de evaluación en función del valor de truncation_percentage que especificó para X.





> Directiva de selección de truncamiento

TruncationSelectionPolicy

Imagine que la métrica principal es la precisión del modelo.

Cuando se registra la precisión de la quinta prueba, la métrica no debe estar en el peor 20 % de las pruebas hasta el momento.

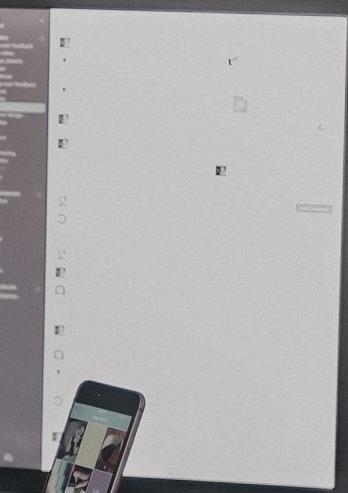
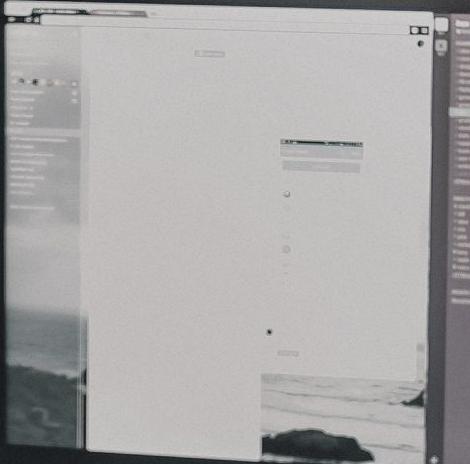
En este caso, el 20 % se traduce en una prueba. En otras palabras, si la quinta prueba no es el modelo con peor rendimiento hasta el momento, el trabajo de barrido continuará.

Si la quinta prueba tiene la puntuación de precisión más baja de todos los ensayos hasta el momento, el trabajo de barrido se detendrá.





```
/* Ejemplo de uso de la función map() */
const array = [1, 2, 3, 4, 5];
const resultado = array.map(function(elemento) {
    return elemento * 2;
});
console.log(resultado);
```



Vamos a la practica!!!



> Uso de un trabajo de barrido para el ajuste de hiperparámetros

1. Configuración del entorno: Asegúrate de tener el entorno y las credenciales configuradas.
2. Definir el trabajo base: Configura el trabajo base que servirá de plantilla.
3. Definir los valores de los hiperparámetros: Utiliza Choice para definir los valores a probar.
4. Configurar el trabajo de barrido: Especifica el clúster, algoritmo de muestreo, métrica y objetivo.
5. Establecer el nombre del experimento: Asigna un nombre al experimento.
6. Definir los límites del barrido: Establece límites en términos de ejecuciones, concurrencia y tiempo.
7. Enviar el trabajo de barrido: Envía el trabajo configurado a Azure ML.





> **Paso 1: Configuración del entorno**

Primero, asegúrate de tener instaladas las librerías necesarias y configurada tu cuenta de Azure.

```
pip install azure-ai-ml
```





> Paso 2: Definir el trabajo base

Define un trabajo base que incluya la configuración del modelo y los hiperparámetros.

```
from azure.ai.ml import command

# Configura el trabajo base
job = command(
    code=".src", # Directorio que contiene el código fuente
    command="python train.py --regularization ${{inputs.reg_rate}}", # Comando para
ejecutar el script con el hiperparámetro
    inputs={
        "reg_rate": 0.01, # Valor por defecto del hiperparámetro
    },
    environment="AzureML-sklearn-0.24-ubuntu18.04-py37-cpu@latest", # Entorno de ejecución
    compute="aml-cluster", # Clúster de computación
)
```





> Paso 3: Definir los valores de los hiperparámetros

Utiliza la clase Choice para definir los valores de los hiperparámetros que se van a probar en el barrido.

```
from azure.ai.ml.sweep import Choice  
  
# Configura el trabajo de barrido con los valores de hiperparámetros  
command_job_for_sweep = job(  
    reg_rate=Choice(values=[0.01, 0.1, 1]), # Conjunto de valores para la tasa de  
    regularización  
)
```





> Paso 4: Configurar el trabajo de barrido

Configura el trabajo de barrido especificando el clúster de computación, el algoritmo de muestreo, la métrica principal y el objetivo del barrido.

```
# Aplica los parámetros de barrido para obtener el trabajo de barrido
sweep_job = command_job_for_sweep.sweep(
    compute="aml-cluster", # Clúster de computación
    sampling_algorithm="grid", # Algoritmo de muestreo (barrido exhaustivo)
    primary_metric="Accuracy", # Métrica principal
    goal="Maximize", # Objetivo de maximizar la métrica principal
)
```





> Paso 5: Establecer el nombre del experimento

Asigna un nombre al experimento para identificar el trabajo de barrido en Azure ML.

```
# Establece el nombre del experimento del trabajo de barrido  
sweep_job.experiment_name = "sweep-example"
```





> Paso 6: Definir los límites del barrido

Establece los límites para el barrido, como el número máximo de ejecuciones, el número máximo de ejecuciones concurrentes y el tiempo máximo permitido.

```
# Define los límites para este barrido
sweep_job.set_limits(
    max_total_trials=4, # Número máximo de ejecuciones del barrido
    max_concurrent_trials=2, # Número máximo de ejecuciones concurrentes
    timeout=7200 # Tiempo máximo permitido (en segundos) para el barrido
)
```





> Paso 7: Enviar el trabajo de barrido

Envía el trabajo de barrido a Azure Machine Learning para su ejecución.

```
from azure.ai.ml import MLClient
from azure.identity import DefaultAzureCredential

# Instancia de MLClient para interactuar con Azure ML
ml_client = MLClient(
    DefaultAzureCredential(),
    subscription_id="tu-subscription-id",
    resource_group_name="tu-resource-group",
    workspace_name="tu-workspace"
)

# Envía el trabajo de barrido a Azure ML para su ejecución
returned_sweep_job = ml_client.create_or_update(sweep_job)
.....
```



Ejercicio: Ejecución de un trabajo de barrido

>_





Microsoft Azure

Search resources, services, and docs (G+/)



Azure services



Create a
resource



Azure OpenAI



Azure AI
services



Azure Machine
Learning



Subscriptions



Azure Lab
Services



Resource
groups



Storage
accounts



Virtual
machines



More services

Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help

Requesting a Cloud Shell.**Succeeded**.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

alejandro [~]\$



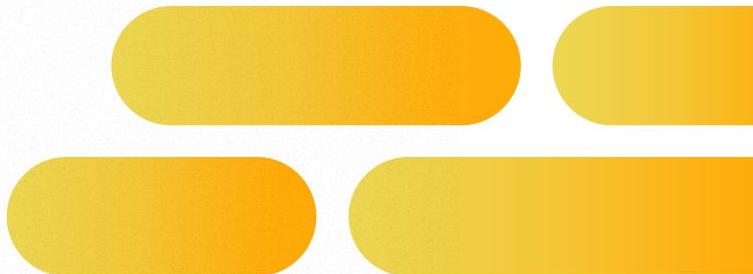


```
alejandro [ ~ ]$ git clone https://github.com/MicrosoftLearning/mslearn-azure-ml.git azure-ml-labs~
Cloning into 'azure-ml-labs~'...
remote: Enumerating objects: 1311, done.
remote: Counting objects: 100% (355/355), done.
remote: Compressing objects: 100% (99/99), done.
remote: Total 1311 (delta 289), reused 257 (delta 256), pack-reused 956
Receiving objects: 100% (1311/1311), 12.77 MiB | 12.51 MiB/s, done.
Resolving deltas: 100% (922/922), done.
alejandro [ ~ ]$ []
```

.....
.....
.....
.....

EdwinWenink Sync metric names in lab 9 dcc8611 · 8 months ago 

Name	Last commit message	Last commit date
 ..		
 data	update numbering	last year
 Hyperparameter tuning.ipynb	Sync metric names in lab 9	8 months ago
 Run a pipeline job.ipynb	update connect to workspace	last year
 setup.sh	remove asia as region	last year



```
remote: Total 1311 (delta 289), reused 257 (delta 256), pack-reused 956
```

```
Receiving objects: 100% (1311/1311), 12.77 MiB | 12.51 MiB/s, done.
```

```
Resolving deltas: 100% (922/922), done.
```

```
alejandro [ ~ ]$ cd azure-ml-labs/Labs/09
```

```
./setup.sh
```

```
Register the Machine Learning resource provider:
```

```
[Survey] Help us improve Azure CLI by sharing your experience. This survey should take about 5 minutes. Run  
ft.com/fwlink/?linkid=2203309
```

```
Create a resource group and set as default:
```

```
[  
  "id": "/subscriptions/4e4f15fc-cca7-492a-935d-d9171a888c0d/resourceGroups/rg-dp100-12007b1022d224a049a",  
  "location": "northeurope",  
  "managedBy": null,  
  "name": "rg-dp100-12007b1022d224a049a",
```

```
.....
```

Code Blame Executable File · 40 lines (32 loc) · 1.5 KB

```
3  # Create random string
4  guid=$(cat /proc/sys/kernel/random/uuid)
5  suffix=${guid//[-]/}
6  suffix=${suffix:0:18}
7
8  # Set the necessary variables
9  RESOURCE_GROUP="rg-dp100-1${suffix}"
10 RESOURCE_PROVIDER="Microsoft.MachineLearning"
11 REGIONS=("eastus" "westus" "centralus" "northeurope" "westeurope")
12 RANDOM_REGION=${REGIONS[$RANDOM % ${#REGIONS[@]}]}
13 WORKSPACE_NAME="mlw-dp100-1${suffix}"
14 COMPUTE_INSTANCE="ci${suffix}"
15 COMPUTE_CLUSTER="aml-cluster"
16
17 # Register the Azure Machine Learning resource provider in the subscription
18 echo "Register the Machine Learning resource provider:"
19 az provider register --namespace $RESOURCE_PROVIDER
20
21 # Create the resource group and workspace and set to default
22 echo "Create a resource group and set as default:"
23 az group create --name $RESOURCE_GROUP --location $RANDOM_REGION
24 az configure --defaults group=$RESOURCE_GROUP
25
26 echo "Create an Azure Machine Learning workspace:"
27 az ml workspace create --name $WORKSPACE_NAME
28 az configure --defaults workspace=$WORKSPACE_NAME
29
30 # Create compute instance
31 echo "Creating a compute instance with name: " $COMPUTE_INSTANCE
32 az ml compute create --name ${COMPUTE_INSTANCE} --size STANDARD_DS11_V2 --type ComputeInstance
33
34 # Create compute cluster
35 echo "Creating a compute cluster with name: " $COMPUTE_CLUSTER
36 az ml compute create --name ${COMPUTE_CLUSTER} --size STANDARD_DS11_V2 --max-instances 2 --type AmlCompute
37
38 # Create data assets
39 echo "Create training data asset:"
40 az ml data create --type uri_file --name "diabetes-data" --path ./data/diabetes.csv
```

Azure Machine Lea... <

Default Directory

[+ Create](#) [Recently deleted](#) ...

Filter for any field...

Name ↑↓

 handson_AutoML	...
 mlw-dp100-l2007b1022d224a049a	...
 ws-cc-ml	...

mlw-dp100-l2007b1022d224a049a ⭐ ...

Azure Machine Learning workspace

 ⌘ ⌘ ⌘[Download config.json](#) [Delete](#)

JSON

Overview

- [Activity log](#)
- [Access control \(IAM\)](#)
- [Tags](#)
- [Diagnose and solve problems](#)
- [Events](#)
- [Settings](#)
- [Monitoring](#)
- [Automation](#)
- [Support + troubleshooting](#)

Essentials

Resource group
[rg-dp100-l2007b1022d224a049a](#)

Location
North Europe

Subscription
[Suscripción de Azure 1](#)

Storage
[mlwdp100storage391bb6020](#)

Studio web URL
<https://ml.azure.com?tid=500f314c-66f1-4fa8-9741-e645a10a0d52...>

Container Registry
...

Key Vault
[mlwdp100keyvaulted8ba16f](#)

Application Insights
[mlwdp100insights3d3ad487](#)

MLflow tracking URI
[azureml://northeurope.api.azureml.ms/mlflow/v1.0/subscriptions/4...](#)

JSON View

Studio web URL

<https://ml.azure.com?tid=500f314c-66f1-4fa8-9741-e645a10a0d52...> 

<https://ml.azure.com?tid=500f314c-66f1-4fa8-9741-e645a10a0d52&wsid=/subscriptions/4e4f15fc-cca7-492a-935d-d9171a888c0d/resourcegroups/rg-dp100-l2007b1022d224a049a/providers/Microsoft.MachineLearningServices/workspaces/mlw-dp100-l2007b1022d224a049a>

Application Insights

≡
← All workspaces

Home
Model catalog

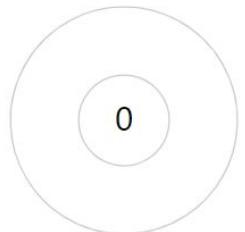
Authoring
Notebooks
Automated ML
Designer
Prompt flow

Assets
Data
Jobs
Components
Pipelines
Environments
Models
Endpoints

Default Directory > mlw-dp100-l2007b1022d224a049a > Compute > aml-cluster

aml-cluster

Cluster node status



- Idle
- Leaving
- Preparing
- Running

Cluster state

Allocation state
 Succeeded (0 nodes)

Allocation state transition time
5/17/2024, 3:15:07 PM

Created on
5/17/2024, 3:14:29 PM
Current node count
0

Attributes

Compute name
aml-cluster
Resource ID
--
Compute type
Machine Learning compute
Subscription ID
4e4f15fc-cca7-492a-935d-d9171a888c0d

Resource properties

Virtual machine size
Standard_DS11_v2 (2 cores, 14 GB RAM, 28 GB disk)
Processing unit
 - Memory optimized
Estimated cost
\$0.18/hr per node
OS Type
Linux

[All workspaces](#)[Home](#)[Model catalog](#)[Authoring](#)[Notebooks](#)[Automated ML](#)[Designer](#)[Prompt flow](#)[Assets](#)[Data](#)[Jobs](#)[Components](#)[Pipelines](#)[Environments](#)[Models](#)[Feedback](#)

Default Directory > mlw-dp100-l2007b1022d224a049a > Compute > ci2007b1022d224a049a

ci2007b1022d224a049a

Resource properties

Status

Running

Last operation

Created at May 17, 2024 3:12 PM: Succeeded

Virtual machine size

Standard_DS11_v2 (2 cores, 14 GB RAM, 28 GB disk)

Processing unit[CPU - Memory optimized](#)**Estimated cost**

\$0.18/hr (when running)

Additional data storage

--

Applications[JupyterLab](#) [Jupyter](#) [VS Code \(Web\)](#) PREVIEW [VS Code \(Desktop\)](#) PREVIEW ...**Created on**

5/17/2024, 3:12:22 PM

SSH access

Disabled

Tags

No tags

Managed identity

No managed identities

Schedules

No schedules

Custom applications

No custom applications

[All workspaces](#)[Home](#)[Model catalog](#)[Authoring](#)[Notebooks](#)[Automated ML](#)[Designer](#)[Prompt flow](#)[Assets](#)[Data](#)[Jobs](#)[Components](#)[Pipelines](#)[Environments](#)[Models](#)[Default Directory](#) > [mlw-dp100-l2007b1022d224a049a](#) > [Compute](#) > [ci2007b1022d224a049a](#)

ci2007b1022d224a049a

Resource properties

Status

Running

Last operation

Created at May 17, 2024 3:12 PM: Succeeded

Virtual machine size

Standard_DS11_v2 (2 cores, 14 GB RAM, 28 GB disk)

Processing unit

CPU - Memory optimized

Estimated cost

\$0.18/hr (when running)

Additional data storage

--

Applications

[JupyterLab](#) [Jupyter](#) [VS Code \(Web\)](#) PREVIEW [VS Code \(Desktop\)](#) PREVIEW

Created on

5/17/2024, 3:12:22 PM

SSH access

Disabled

Tags

No tags

Managed identity

No managed identities

Schedules

No schedules

Custom applications

No custom applications

[Terminal](#)[Notebook](#)

Default Directory > mlw-dp100-l2007b1022d224a049a > Notebooks

1:cj2007b1022d224a04c ×



Compute: ci2007b1022d224a049a - Running ▾

Welcome to the Azure Machine Learning terminal

Enter "git clone [url]" to clone a repo

Enter "git --help" to learn about Git CLI. To learn more about integrating Git with the Azure Machine Learning terminal, navigate here:

<https://learn.microsoft.com/en-us/azure/machine-learning/concept-train-model-git-integration#clone-git-repositories-into-your-workspace-file-system>

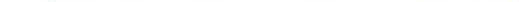
Enter "az ml --help" to learn about Azure ML CLI v2

Note: Use "az login --identity" instead of "az login" to avoid device code authentication.

```
(azureml_py38) azureuser@ci2007b1022d224a049a:~/cloudfiles/code/Users/ayuda$ pip uninstall azure-ai-ml  
WARNING: Skipping azure-ai-ml as it is not installed.
```

```
(azureml_py38) azureuser@ci2007b1022d224a049a:~/cloudfiles/code/Users/ayuda$ pip install azure-ai-ml  
Collecting azure-ai-ml
```

Downloading azure_ai_ml-1.16.0-py3-none-any.whl (10.5 MB)



```
Requirement already satisfied: opencensus-ext-azure in /anaconda/envs/azureml_py38/lib/python3.8/site-packages (from azure-ai-ml) (1.1.9)
Collecting strictyaml
```

Collecting strictyaml
 Downloading strictyaml-1.7.3-py3-none-any.whl (123 kB)

Requirement already satisfied: typing-extensions in /anaconda/envs/azureml_py38/lib/python3.8/site-packages (from azure-ai-ml) (4.6.0)

Requirement already satisfied: pyyaml>=5.1.0 in /anaconda/envs/azureml_py38/lib/python3.8/site-packages (from azure-ai-ml) (6.0)

Requirement already satisfied: azure-common>=1.1 in /anaconda/envs/azureml_py38/lib/python3.8/site-packages (from azure-ai-ml) (1.1.28)

Collecting azure-storage-file-share

Downloading azure storage file sh

All workspaces

Home

Model catalog

Authoring

Notebooks

Automated ML

Designer

Prompt flow

Assets

Data

Jobs

Components

Pipelines

Environments

Models

Endpoints

Default Directory > mlw-dp100-l2007b1022d224a049a > Notebooks

1:ci2007b1022d224a049x



Compute: ci2007b1022d224a049a - Running



```
.9)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /anaconda/envs/azureml_py38/lib/python3.8/site-packages (from google-auth<3.0dev,>=2.1
4.1->google-api-core<3.0.0,>=1.0.0; python_version >= "3.6"->opencensus<1.0.0,>=0.11.2->opencensus-ext-azure->azure-ai-ml) (0.3.0)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /anaconda/envs/azureml_py38/lib/python3.8/site-packages (from google-auth<3.0dev,>=2.
14.1->google-api-core<3.0.0,>=1.0.0; python_version >= "3.6"->opencensus<1.0.0,>=0.11.2->opencensus-ext-azure->azure-ai-ml) (5.3.0)
Requirement already satisfied: pyasn1>=0.1.3 in /anaconda/envs/azureml_py38/lib/python3.8/site-packages (from rsa<5,>=3.1.4; python_version >=
"3.6"->google-auth<3.0dev,>=2.14.1->google-api-core<3.0.0,>=1.0.0; python_version >= "3.6"->opencensus<1.0.0,>=0.11.2->opencensus-ext-azure->
azure-ai-ml) (0.5.0)
ERROR: azure-storage-file-share 12.16.0 has requirement azure-core>=1.28.0, but you'll have azure-core 1.26.4 which is incompatible.
ERROR: pydash 8.0.1 has requirement typing-extensions!=4.6.0,>3.10, but you'll have typing-extensions 4.6.0 which is incompatible.
ERROR: azure-storage-file-datalake 12.15.0 has requirement azure-core>=1.28.0, but you'll have azure-core 1.26.4 which is incompatible.
ERROR: azure-storage-file-datalake 12.15.0 has requirement azure-storage-blob>=12.20.0, but you'll have azure-storage-blob 12.13.0 which is in
compatible.
Installing collected packages: strictyaml, azure-storage-file-share, pydash, azure-storage-file-datalake, marshmallow, opencensus-ext-logging,
azure-ai-ml
Successfully installed azure-ai-ml-1.16.0 azure-storage-file-datalake-12.15.0 azure-storage-file-share-12.16.0 marshmallow-3.21.2 opencensus-e
xt-logging-0.1.1 pydash-8.0.1 strictyaml-1.7.3
(azureml_py38) azureuser@ci2007b1022d224a049a:~/cloudfiles/code/Users/ayuda$ git clone https://github.com/MicrosoftLearning/mslearn-azure-ml.
git azure-ml-labs
Cloning into 'azure-ml-labs'...
remote: Enumerating objects: 1311, done.
remote: Counting objects: 100% (355/355), done.
remote: Compressing objects: 100% (99/99), done.
remote: Total 1311 (delta 289), reused 257 (delta 256), pack-reused 956
Receiving objects: 100% (1311/1311), 12.77 MiB | 7.61 MiB/s, done.
Resolving deltas: 100% (922/922), done.
Updating files: 100% (99/99), done.
(azureml_py38) azureuser@ci2007b1022d224a049a:~/cloudfiles/code/Users/ayuda$
```

Default Directory > mlw-dp100-
1:ci2007b1022d224a04s X

≡ ? ☰

Welcome to the Azure M...

Enter "git clone [url]"
Enter "git --help" to ...

https://learn...
e-file-system
Enter "az ml --help" to ...

Note: Use "az login ..." to log in.

(azureml_py38) azureus:
WARNING: Skipping azure-ai-ml
(azureml_py38) azureus:
Collecting azure-ai-ml
 Downloading azure_ai...
Requirement already satisfied: strictyaml
Collecting strictyaml
 Downloading strictyaml...
Requirement already satisfied: azure-storage
Requirement already satisfied: azure-storage
Collecting azure-storage
 Downloading azure_sto...

Notebooks

Files Samples

< X CSV + ⌂ ⌂ ⌂

- > Logs
- ▽ Users
- ▽ ayuda
- > azure-ml-labs

[All workspaces](#)[Home](#)[Model catalog](#)[Authoring](#)[Notebooks](#)[Automated ML](#)[Designer](#)[Prompt flow](#)[Assets](#)[Data](#)[Jobs](#)[Components](#)[Pipelines](#)[Environments](#)[Models](#)

Default Directory > mlw-dp100-l2007b1022d224a049a > Notebooks

Notebooks

[Files](#) [Samples](#)

07

08

09

data

[Hyperparameter tuning.ipynb](#) [Run a pipeline job.ipynb](#) [setup.sh](#)

10

11

Policies

typings

_build.yml

1:ci2007b1022d224a049a X

Hyperparameter tunir X



Edit in VS Code

Compute:

ci20...



You need to be authenticated to the compute to use any Azure SDK. Please use the authenticate button to get authentication.

ci2007b1022d224a049a · Kernel idle CPU 0% RAM 2%

Last saved 2 minutes ago

Define the search space

When your command job has completed successfully, you can configure and run a sweep job.

First, you'll need to specify the search space for your hyperparameter. To train three models, each (`0.01`, `0.1`, or `1`), you can define the search space with a `Choice` hyperparameter.

```
1 from azure.ai.ml.sweep import Choice
2
3 command_job_for_sweep = job(
4     reg_rate=Choice(values=[0.01, 0.1, 1]),
5 )
```

Configure and submit the sweep job

1:ci2007b1022d224a049

Hyperparameter tunir



Edit in VS Code

Compute:

ci20...



Python

(i) You need to be authenticated to the compute to use any Azure SDK. Please use the authenticate button to get authenticated.

ci2007b1022d224a049 · Kernel idle CPU 0% RAM 2%

Last saved 3 minutes ago

Python 3 (ipykernel)

Python 3.10 - SDK v2

Python 3.8 - AzureML

Python 3.8 - Pytorch and
Tensorflow

R

Define the search space

When your command job has completed successfully, you can configure and run a sweep job.

First, you'll need to specify the search space for your hyperparameter. To train three models, each with a different regularization rate (`0.01`, `0.1`, or `1`), you can define the search space with a `Choice` hyperparameter.



1:ci2007b1022d224a049a

*Hyperparameter tun



Edit in VS Code

Compute:

ci2007b1022d224a049a - Running



Python 3.8 - AzureML



» ⓘ Successfully authenticated to the compute. You can now use any Azure SDK.

ci2007b1022d224a049a · Kernel idle CPU 20% RAM 3%

Last saved a few seconds ago

Python 3.8 - AzureML

Run the following cell to submit the sweep job.



```
1  returned_sweep_job = ml_client.create_or_update(sweep_job)
2  aml_url = returned_sweep_job.studio_url
3  print("Monitor your job at", aml_url)
```

[9] ✓ 2 sec

... Monitor your job at https://ml.azure.com/runs/red_shoe_c3bwvp6g0b?wsid=/subscriptions/4e4f15fc-cca7-492a-935d-d9171a888c0d/resourcegroups/rg-dp100-l2007b1022d224a049a/workspaces/mlw-dp100-l2007b1022d224a049a&tid=500f314c-66f1-4fa8-9741-e645a10a0d52

When the job is completed, navigate to the job overview. The **Trials** tab will show all models that have been trained and how the `Accuracy` score differs for each regularization rate value you tried.

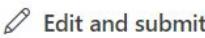


diabetes-train-mlflow



Running

Overview Trials Metrics Outputs + logs Code



Properties

Status
 Running
Created on
May 17, 2024 3:54 PM

Start time
May 17, 2024 3:54 PM

Compute target
aml-cluster

Name
red_shoe_c3bwvp6g0b

Created by
Alejandro Soria

Job type
Sweep
Experiment
[sweep-diabetes](#)

Registered models
None

[See all properties](#)
 Raw JSON

[See YAML job definition](#)
 Job YAML

Parameter sampling

Sampling policy name
GRID
Parameter space
`{"reg_rate":["choice",[[0.01,0.1,1]]]}`

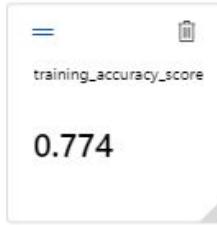
Early termination policy

Early termination policy
Default

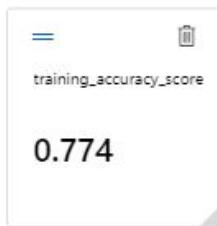
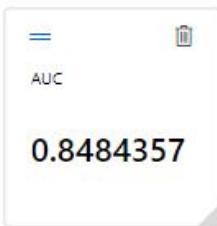
Properties
`{}`

Primary metric

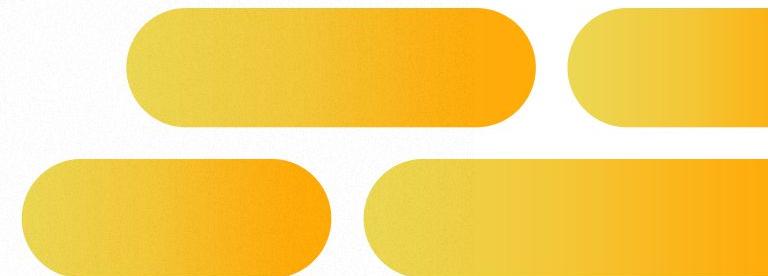
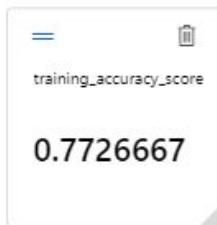
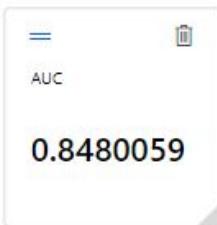
Select metrics



▼ (diabetes-train-mlflow_1) (2)



▼ (diabetes-train-mlflow_2) (2)



Primary metric

Primary metric name

training_accuracy_score

Primary metric goal

maximize

Best trial

red_shoe_c3bwvp6g0b_1

diabetes-train-mlflow_1 Completed

Overview Metrics Images Child jobs Outputs + logs Code Explanations (preview) Fairness (preview) Monitoring

Properties

Status	Created by
Completed	Alejandro Soria
Created on	Job type
May 17, 2024 3:54 PM	Command
Start time	Experiment
May 17, 2024 3:59 PM	sweep-diabetes
Duration	Environment
17.61s	AzureML-sklearn-0.24-ubuntu18.04-py37-cpu:49
Compute duration	Registered models
17.61s	None
Name	See all properties
red_shoe_c3bwvp6g0b_1	
Command	See YAML job definition
python train.py --training_data \${{inputs.diabetes_data}} --reg_rate \${{search_space.reg_rate}}	

Inputs

Input name: diabetes_data
Data asset: diabetes-data:1
Asset URI: azureml:diabetes-data:1

Tags

hyperparameters : {"reg_rate": 0.1} model_type : LogisticRegression

Params

Regularization rate : 0.1

Metrics

AUC	training_accuracy_score
0.8484357	0.774



códigofacilito

Optimización del entrenamiento de modelos con Azure Machine Learning

Ing. Cinthya Cabanzo





> Agenda

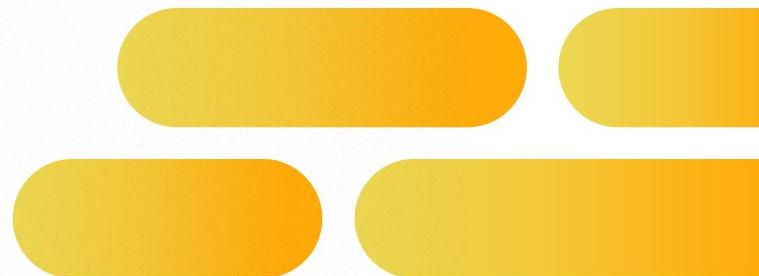
- > Introducción
- > Crear componentes
- > Crear una canalización
- > Ejecución de un trabajo de canalización
- > Ejercicio: Ejecución de un trabajo de canalización





Introducción

- En Azure Machine Learning, puede experimentar en cuadernos y entrenar modelos de aprendizaje automático **ejecutando scripts como trabajos (jobs)**.
- Para implementar una solución eficaz de operaciones de aprendizaje automático (MLOps) en Azure, se **agrupan tareas en canalizaciones(pipelines)**, separando el proceso global en **tareas individuales**, llamadas **componentes**.



Crear componentes

Los componentes sirven para:

- Para compilar una canalización.
- Para compartir código listo para usar.

Un componente consta de tres partes: **Metadatos**: incluye el nombre, la versión, etc. del componente. **Interfaz**: incluye los parámetros de entrada esperados (como un conjunto de datos o hiperparámetros) y la salida esperada (como métricas y artefactos).

Comando, código y entorno: especifica cómo ejecutar el código.

Para crear un componente, necesita dos archivos:

- **Script** que contiene el flujo de trabajo que desea ejecutar.
- Un archivo **YAML** para definir los metadatos, la interfaz y el comando, el código y el entorno del componente.



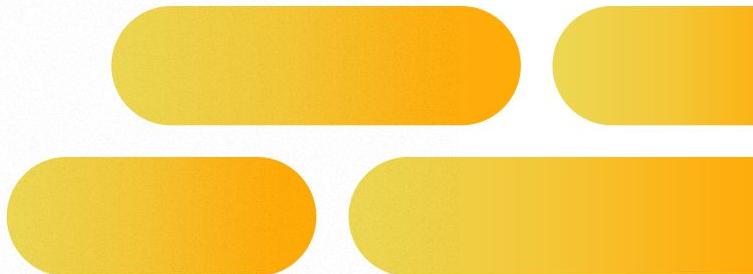
>Create components



prep.py
normaliza
datos



Componente



yml

```
$schema: https://azuremlschemas.azureedge.net/latest/commandComponent.schema.json
name: prep_data
display_name: Prepare training data
version: 1
type: command
inputs:
    input_data:
        type: uri_file
outputs:
    output_data:
        type: uri_file
code: ./src
environment: azureml:AzureML-sklearn-0.24-ubuntu18.04-py37-cpu@latest
command: >-
    python prep.py
    --input_data ${{inputs.input_data}}
    --output_data ${{outputs.output_data}}
```

YML para crear componentes (explicado)

```
# Esquema del componente de comando, utilizado para definir los detalles del componente
$schema: https://azuremlschemas.azureedge.net/latest/commandComponent.schema.json

# Nombre del componente. Este nombre debe ser único dentro del espacio de trabajo de Azure Machine Learning.
name: prep_data

# Nombre que se muestra para el componente en la interfaz de usuario de Azure Machine Learning.
display_name: Prepare training data

# Versión del componente. Esto permite tener varias versiones del mismo componente.
version: 1

# Tipo de componente, que en este caso es un comando que se ejecutará.
type: command
```



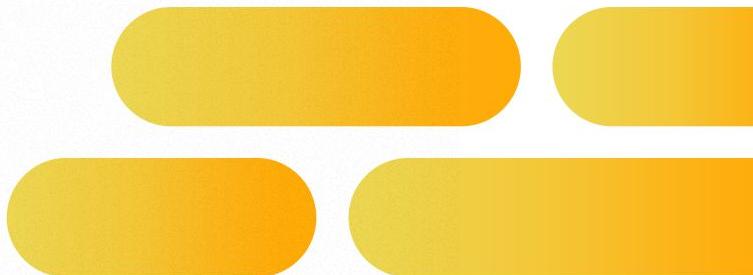
YML para crear componentes

```
# Definición de las entradas del componente.  
inputs:  
    # Definición de una entrada llamada 'input_data'.  
    input_data:  
        # Tipo de la entrada. En este caso, se espera una URI que apunta a un archivo.  
        type: uri_file  
  
# Definición de las salidas del componente.  
outputs:  
    # Definición de una salida llamada 'output_data'.  
    output_data:  
        # Tipo de la salida. En este caso, se espera una URI que apunta a un archivo.  
        type: uri_file  
  
# Ruta al directorio que contiene el código fuente que se va a ejecutar.  
code: ./src
```



YML para crear componentes

```
# Entorno de ejecución que especifica la imagen de Docker y el entorno de Python que se utilizarán.  
environment: azureml:AzureML-sklearn-0.24-ubuntu18.04-py37-cpu@latest  
  
# Comando que se ejecutará cuando se ejecute el componente.  
# El comando ejecuta un script Python llamado 'prep.py' y pasa las entradas y salidas como argumentos.  
command: >-  
    python prep.py  
    --input_data ${{inputs.input_data}}  
    --output_data ${{outputs.output_data}}
```



YML para cargar componentes

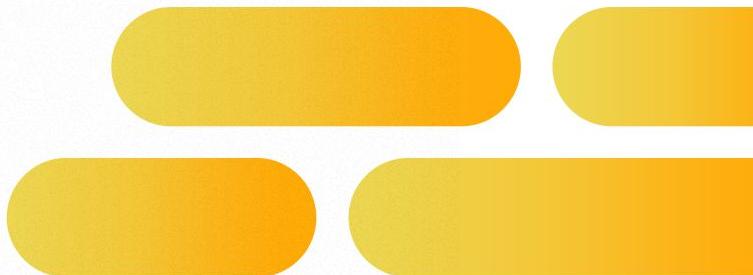
```
# Importa la función load_component desde el módulo azure.ai.ml.  
# Esta función se utiliza para cargar componentes definidos en archivos YAML en Azure  
Machine Learning.  
from azure.ai.ml import load_component  
  
# Define la variable parent_dir como una cadena vacía.  
# Esta variable se utilizará como el directorio base donde se encuentra el archivo  
YAML del componente.  
parent_dir = ""  
  
# Llama a la función load_component para cargar el componente definido en el archivo  
prep.yml.  
# La función toma como argumento la ruta al archivo YAML que describe el componente.  
# La ruta al archivo se construye concatenando parent_dir con "./prep.yml".  
# El resultado de la función load_component se almacena en la variable  
loaded_component_prep.  
loaded_component_prep = load_component(source=parent_dir + "./prep.yml")
```



YML para registrar componentes

```
# Asume que ml_client es una instancia del cliente de Azure Machine Learning
# Asume que prepare_data_component es una definición de componente que deseas crear o
actualizar

# Llama al método components.create_or_update del cliente de Azure Machine Learning
(ml_client)
# Este método crea un nuevo componente o actualiza uno existente basado en la
definición proporcionada
# El componente que se crea o actualiza se guarda en la variable prep
prep = ml_client.components.create_or_update(loader_component_prep )
```





Crear una canalización





Crear una canalización

```
from azure.ai.ml.dsl import pipeline

@pipeline()
def pipeline_function_name(pipeline_job_input):
    prep_data = loaded_component_prep(input_data=pipeline_job_input)
    train_model = loaded_component_train(training_data=prep_data.outputs.output_data)

    return {
        "pipeline_job_transformed_data": prep_data.outputs.output_data,
        "pipeline_job_trained_model": train_model.outputs.model_output,
    }
```

El resultado de ejecutar la función `@pipeline()` es un archivo YAML

Crear una canalización

```
# La función pipeline se utiliza para definir una canalización de Azure Machine Learning.
from azure.ai.ml.dsl import pipeline

# Define una función de canalización utilizando el decorador @pipeline.
# El decorador @pipeline indica que esta función es una canalización de Azure Machine Learning.
@pipeline()
def pipeline_function_name(pipeline_job_input):
    # Ejecuta el componente de preparación de datos (loaded_component_prep) con la entrada especificada
    # (pipeline_job_input).
    # Asigna la salida del componente a la variable prep_data.
    prep_data = loaded_component_prep(input_data=pipeline_job_input)

    # Ejecuta el componente de entrenamiento de modelo (loaded_component_train) con los datos preparados por el
    # componente anterior.
    # Utiliza la salida del componente prep_data como entrada para el componente de entrenamiento.
    # Asigna la salida del componente de entrenamiento a la variable train_model.
    train_model = loaded_component_train(training_data=prep_data.outputs.output_data)

    # Retorna un diccionario con las salidas de la canalización.
    # Incluye la salida del componente de preparación de datos y la salida del componente de entrenamiento de mode
    return {
        "pipeline_job_transformed_data": prep_data.outputs.output_data,
        "pipeline_job_trained_model": train_model.outputs.model_output,
    }
```



Resultado de la canalización

yml

```
display_name: pipeline_function_name
type: pipeline
inputs:
  pipeline_job_input:
    type: uri_file
    path: azureml:data:1
outputs:
  pipeline_job_transformed_data: null
  pipeline_job_trained_model: null
jobs:
  prep_data:
    type: command
    inputs:
      input_data:
        path: ${parent.inputs.pipeline_job_input}
    outputs:
      output_data: ${parent.outputs.pipeline_job_transformed_data}
  train_model:
    type: command
    inputs:
      input_data:
        path: ${parent.outputs.pipeline_job_transformed_data}
    outputs:
      output_model: ${parent.outputs.pipeline_job_trained_model}
tags: {}
properties: {}
settings: {}
```



Resultado de una canalización (explicado)

```
# Define el esquema utilizado para validar la definición del componente.  
$schema: https://azuremlschemas.azureedge.net/latest/pipelineComponent.schema.json  
  
# Nombre de la canalización.  
display_name: pipeline_function_name  
  
# Tipo de componente, en este caso, una canalización.  
type: pipeline  
  
# Definición de las entradas de la canalización.  
inputs:  
    # Define una entrada llamada pipeline_job_input de tipo uri_file.  
    pipeline_job_input:  
        type: uri_file  
        # Especifica la ruta del archivo de entrada.  
        path: azureml:data:1  
  
# Definición de las salidas de la canalización.  
outputs:  
    # Define una salida llamada pipeline_job_transformed_data, inicializada como nula.  
    pipeline_job_transformed_data: null  
    # Define una salida llamada pipeline_job_trained_model, inicializada como nula.  
    pipeline_job_trained_model: null
```



```
# Definición de los trabajos (jobs) que se ejecutarán en la canalización.
jobs:
    # Definición del trabajo de preparación de datos.
    prep_data:
        # Tipo de trabajo, en este caso un comando.
        type: command
        # Definición de las entradas para el trabajo prep_data.
        inputs:
            # Define una entrada llamada input_data.
            input_data:
                # Especifica la ruta de la entrada como la entrada de la canalización.
                path: ${parent.inputs.pipeline_job_input}
        # Definición de las salidas para el trabajo prep_data.
        outputs:
            # Define una salida llamada output_data.
            output_data: ${parent.outputs.pipeline_job_transformed_data}

# Definición del trabajo de entrenamiento de modelo.
train_model:
    # Tipo de trabajo, en este caso un comando.
    type: command
    # Definición de las entradas para el trabajo train_model.
    inputs:
        # Define una entrada llamada input_data.
        input_data:
            # Especifica la ruta de la entrada como la salida del trabajo prep_data.
            path: ${parent.outputs.pipeline_job_transformed_data}
    # Definición de las salidas para el trabajo train_model.
    outputs:
        # Define una salida llamada output_model.
        output_model: ${parent.outputs.pipeline_job_trained_model}
```



> Ejecución de un trabajo de canalización

Python

```
# submit job to workspace
pipeline_job = ml_client.jobs.create_or_update(
    pipeline_job, experiment_name="pipeline_job"
)
```





> crear una programación que se active cada minuto

Python

```
from azure.ai.ml.entities import RecurrenceTrigger  
  
schedule_name = "run_every_minute"  
  
recurrence_trigger = RecurrenceTrigger(  
    frequency="minute",  
    interval=1,  
)
```





> Programar una canalizacion

Python

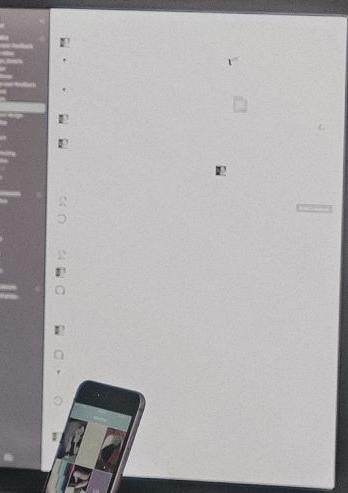
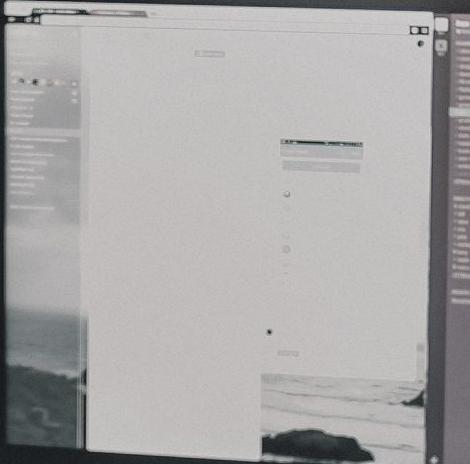
```
from azure.ai.ml.entities import JobSchedule

job_schedule = JobSchedule(
    name=schedule_name, trigger=recurrence_trigger, create_job=pipeline_job
)

job_schedule = ml_client.schedules.begin_create_or_update(
    schedule=job_schedule
).result()
```



```
/* Ejemplo de uso de la función map() */
const array = [1, 2, 3, 4, 5];
const resultado = array.map(function(elemento) {
    return elemento * 2;
});
console.log(resultado);
```



Vamos a la practica!!!



> Ejercicio: Ejecución de un trabajo de canalización

The screenshot shows a workspace interface for Azure Machine Learning. The left sidebar lists notebooks and files, including 'Logs', 'Users', 'ayuda', 'azure-ml-labs' (which contains 'Instructions', 'Labs', '02', '03', '04', '05', '06', '07', '08', '09', 'data', 'Hyperparameter tuning.ipynb', 'Run a pipeline job.ipynb', and 'setup.sh'), and '10'. The main area displays a notebook titled 'ci2007b1022d224a049a - Kernel busy CPU100% RAM 4%' with the status 'Last saved a few seconds ago'. The notebook content includes a section titled 'Run scripts as a pipeline job' which explains what a pipeline is and how to verify the installation of the 'azure-ai-ml' package. A code cell shows the command 'pip show azure-ai-ml' and its output, which confirms the package is installed with version 1.16.0. Below this, there is a section titled 'Connect to your workspace' with a note about required SDK packages.

Default Directory > mlw-dp100-l2007b1022d224a049a > Notebooks

Notebooks

Files Samples

Logs, Users, ayuda, azure-ml-labs (Instructions, Labs, 02, 03, 04, 05, 06, 07, 08, 09, data), Hyperparameter tuning.ipynb, Run a pipeline job.ipynb, setup.sh, 10

1:ci2007b1022d224a049a > Hyperparameter tuning.ipynb > *Run a pipeline job.ipynb

Edit in VS Code Compute: ci2007b1022d224a049a - Running Python 3.8 - AzureML Python 3.8 - AzureML

ci2007b1022d224a049a - Kernel busy CPU100% RAM 4% Last saved a few seconds ago

Run scripts as a pipeline job

A pipeline allows you to group multiple steps into one workflow. You can build a pipeline with components. Each component reflects a Python script to run. A component is defined in a YAML file which specifies the script and how to run it.

Before you start

You'll need the latest version of the `azure-ai-ml` package to run the code in this notebook. Run the cell below to verify that it is installed.

Note: If the `azure-ai-ml` package is not installed, run `pip install azure-ai-ml` to install it.

```
1 pip show azure-ai-ml
[1]: ✓ 4sec
...
Name: azure-ai-ml
Version: 1.16.0
Summary: Microsoft Azure Machine Learning Client Library for Python
Home-page: https://github.com/Azure/azure-sdk-for-python
Author: Microsoft Corporation
Author-email: azuresdkengsysadmins@microsoft.com
License: MIT License
Location: /anaconda/envs/azureml_py38/lib/python3.8/site-packages
Requires: strictyaml, marshmallow, mrest, azure-core, azure-storage-file-datalake, tqdm, pyyaml, opencensus-ext-logging, jsonschema, pydash, isodate, azure-mgmt-core, opencensus-ext-azure, azure-storage-blob, colorama, azure-storage-file-share, azure-common, typing-extensions, pyjwt
Required-by:
```

Note: you may need to restart the kernel to use updated packages.

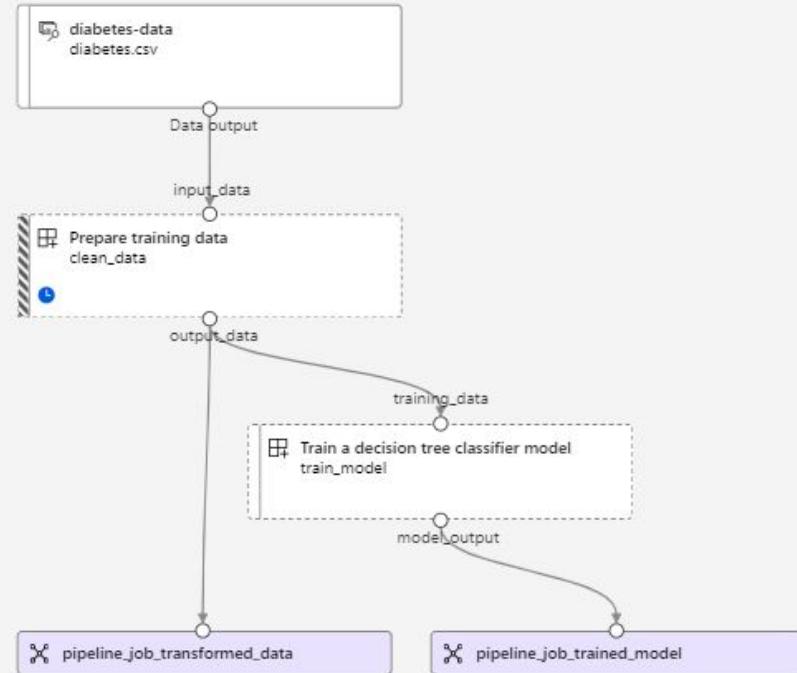
Connect to your workspace

With the required SDK packages installed, now you're ready to connect to your workspace.



>_

Job



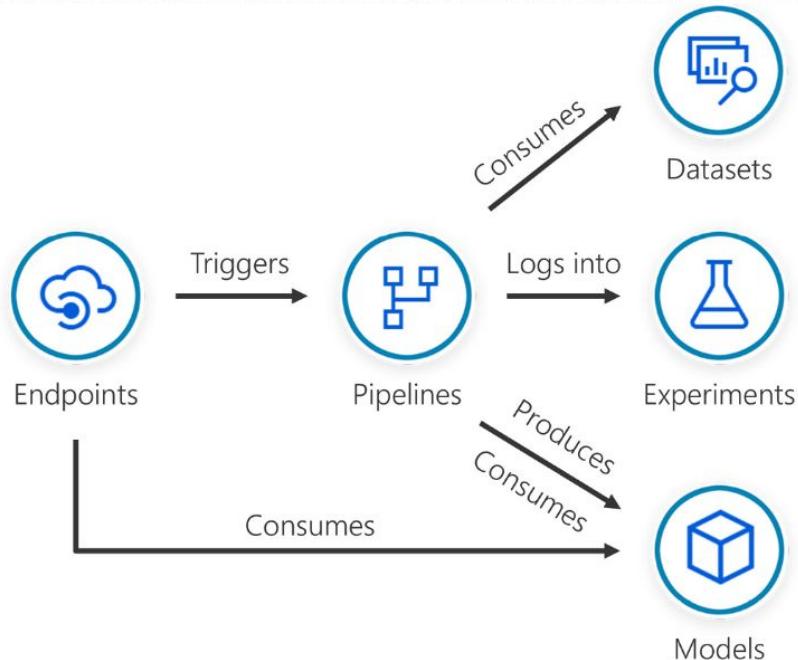


Figure 3.7 – Interconnected assets in Azure ML





>

Ejercicio de tarea:

<https://microsoftlearning.github.io/mslearn-azure-ml/Instructions/10-Compare-models.html>

<https://microsoftlearning.github.io/mslearn-azure-ml/Instructions/10-Log-mlflow-models.html>





```
//life motto  
if (sad( )==true){  
    sad( ).stop( );  
    beAwesome( );  
}
```



- ≡ “La imaginación es más importante que el conocimiento. El conocimiento es limitado y la imaginación circunda el mundo”.
- >_
- >_

“Lo importante es no dejar de hacerse preguntas”.

Albert Einstein

