

ESPECIALIZACIÓN EN MACHINE LEARNING ENGINEER

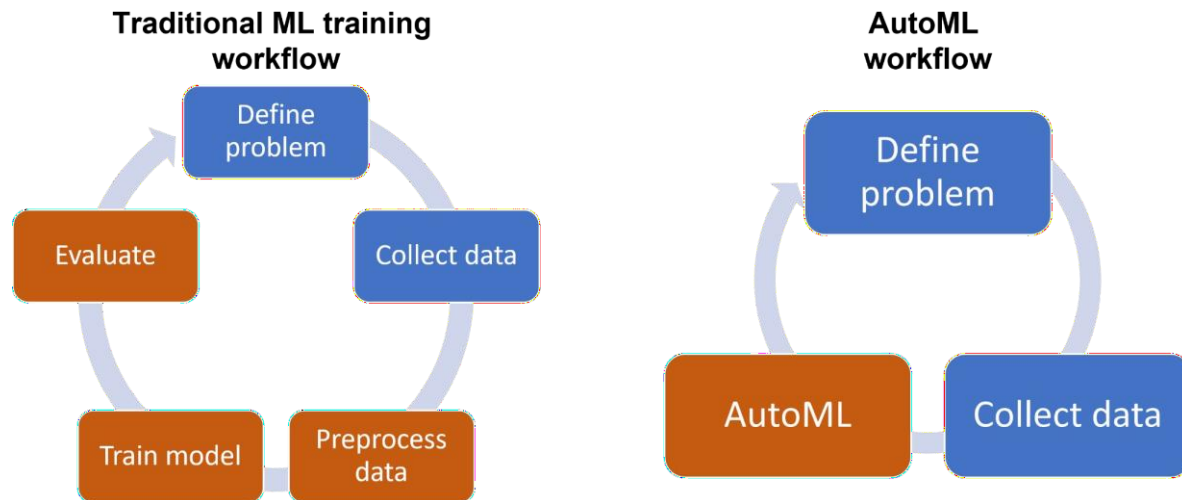
Tema: Fundamentos de MLE

Sesión 4

Docente: Arnaldo Alvarado

¿Qué es AutoML?

El Machine Learning Automatizado (**AutoML**) representa un conjunto de procesos y técnicas que permiten la **automatización de tareas** relacionadas con el aprendizaje automático. Su objetivo es hacer que el Machine Learning sea más accesible, reduciendo o eliminando la necesidad de **expertise especializado en ciertas etapas** del proceso de modelado. Esto incluye desde la preparación de los datos, selección de modelos, hasta la optimización de hiperparámetros.



PyCaret: Conceptos

PyCaret es una **biblioteca de AutoML en Python** que ofrece una interfaz de alto nivel para varias bibliotecas y frameworks de ML, como scikit-learn, XGBoost, LightGBM, y otros. Está diseñada para **automatizar el flujo de trabajo de Machine Learning**, permitiendo a los usuarios pasar de la preparación de datos a la implementación de modelos en pocas líneas de código.

Simplicidad: PyCaret está diseñado para ser **simple y fácil de usar**. Los usuarios pueden realizar tareas complejas de ML con un **mínimo esfuerzo y conocimiento técnico**.

Eficiencia: Permite a los usuarios **experimentar y comparar docenas de modelos** y técnicas de preprocesamiento de forma rápida y eficiente.

Flexible: Aunque es de alto nivel y fácil de usar, PyCaret **no sacrifica la flexibilidad**. Los usuarios pueden **personalizar el pipeline** de ML según sea necesario.

Integrado: Soporta tareas de clasificación, regresión, clustering, detección de anomalías, procesamiento de lenguaje natural (NLP), y más, todo dentro de un marco unificado.

Instalación

```
pip install pycaret
```

PyCaret: Conceptos

Preparación de datos

La calidad de los datos de entrada es **fundamental para el éxito de cualquier proyecto** de Machine Learning. La preparación de los datos implica varios pasos críticos para asegurar que tus **modelos funcionen correctamente** y sean capaces de generar **predicciones precisas**.

Limpieza de datos: Elimina o corrige registros corruptos o inexactos de tus datos. Esto puede incluir el manejo de **valores faltantes**, la **eliminación de duplicados**, o la **corrección de errores** de formato.

Transformación de datos: Convierte los datos a un **formato adecuado** para el modelado. Esto puede implicar la **normalización o estandarización** de las características numéricas, la **codificación de variables categóricas**, y la **transformación de fechas** y otros tipos de datos en formatos útiles.

Selección de features: Identifica las **características más relevantes** para tu problema. Esto puede implicar el **análisis de la correlación** entre diferentes características y la etiqueta objetivo, así como la eliminación de características redundantes o irrelevantes.

División de datos: Divide tus datos en **conjuntos de entrenamiento y prueba** (y, opcionalmente, de validación). Esto es crucial para evaluar el rendimiento de tus modelos de manera objetiva, entrenándolos en un conjunto de datos y probándolos en otro distinto.

PyCaret **simplifica muchos de estos pasos con su función setup**, la cual realiza automáticamente muchas tareas de preprocesamiento y validación de datos, permitiéndote especificar tus requisitos y preferencias a través de parámetros. Sin embargo, es **importante realizar un análisis exploratorio de datos (EDA)** antes de este paso para entender tus datos y determinar las transformaciones y limpiezas necesarias.

PyCaret: AutoML

Paso 1: Cargar dataset

PyCaret ofrece datasets de muestra que pueden ser cargados directamente. Para cargar el dataset de Iris:

```
from pycaret.datasets import get_data
dataset = get_data('iris')
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Si quieres importar tus propios datos, puedes utilizar pandas y luego pasar el DataFrame a PyCaret:

```
import pandas as pd
dataset = pd.read_csv('mi_dataset.csv')
```

PyCaret: AutoML

Paso 2: Configuración del experimento

Para inicializar el entorno en PyCaret y configurar el experimento, usamos `setup()`. Este paso es crucial ya que prepara los datos para el modelado y establece las bases del experimento.

```
from pycaret.classification import setup
exp_clf101 = setup(data          = dataset,
                   target       = 'species',
                   session_id    = 123,
                   train_size    = 0.7,
                   numeric_features = ["sepal_length",
                                     "sepal_width", "petal_length", "petal_width"])
```

<https://pycaret.readthedocs.io/en/stable/api/classification.html>

	Description	Value
0	Session id	123
1	Target	species
2	Target type	Multiclass
3	Target mapping	Iris-setosa: 0, Iris-versicolor: 1, Iris-virginica: 2
4	Original data shape	(150, 5)
5	Transformed data shape	(150, 5)
6	Transformed train set shape	(105, 5)
7	Transformed test set shape	(45, 5)
8	Numeric features	4
9	Preprocess	True
10	Imputation type	simple
11	Numeric imputation	mean
12	Categorical imputation	mode
13	Fold Generator	StratifiedKfold
14	Fold Number	10
15	CPU Jobs	-1
16	Use GPU	False
17	Log Experiment	False
18	Experiment Name	clf-default-name
19	USI	079f

PyCaret: AutoML

Paso 3: Comparación de modelos

Para comparar diferentes modelos y encontrar el de mejor rendimiento según una métrica determinada, utilizamos `compare_models()`.

```
from pycaret.classification import compare_models
best_model = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.9718	0.0000	0.9718	0.9780	0.9712	0.9573	0.9609	0.8210
knn	K Neighbors Classifier	0.9718	0.0000	0.9718	0.9780	0.9712	0.9573	0.9609	0.4020
qda	Quadratic Discriminant Analysis	0.9718	0.0000	0.9718	0.9780	0.9712	0.9573	0.9609	0.0150
lda	Linear Discriminant Analysis	0.9718	0.0000	0.9718	0.9780	0.9712	0.9573	0.9609	0.0150
lightgbm	Light Gradient Boosting Machine	0.9536	0.0000	0.9536	0.9634	0.9528	0.9298	0.9356	0.1000
nb	Naive Bayes	0.9445	0.0000	0.9445	0.9525	0.9438	0.9161	0.9207	0.0110
et	Extra Trees Classifier	0.9445	0.0000	0.9445	0.9586	0.9426	0.9161	0.9246	0.0530
gbc	Gradient Boosting Classifier	0.9355	0.0000	0.9355	0.9416	0.9325	0.9023	0.9083	0.0990
dt	Decision Tree Classifier	0.9264	0.0000	0.9264	0.9502	0.9201	0.8886	0.9040	0.0100
rf	Random Forest Classifier	0.9264	0.0000	0.9264	0.9343	0.9232	0.8886	0.8956	0.0720
ada	Ada Boost Classifier	0.9155	0.0000	0.9155	0.9401	0.9097	0.8720	0.8873	0.0370
ridge	Ridge Classifier	0.8227	0.0000	0.8227	0.8437	0.8186	0.7320	0.7454	0.0110
svm	SVM - Linear Kernel	0.7618	0.0000	0.7618	0.6655	0.6888	0.6333	0.7048	0.0150
dummy	Dummy Classifier	0.2864	0.0000	0.2864	0.0822	0.1277	0.0000	0.0000	0.0100

PyCaret: AutoML

Paso 4: Creación de un modelo específico

Si tienes interés en un modelo específico, puedes crearlo directamente. Por ejemplo, para entrenar un modelo de Árbol de Decisión:

```
from pycaret.classification import
create_model
dt = create_model('lr')
```

'lr' se refiere a Logistic Regression. Cada tipo de modelo tiene una identificación específica en PyCaret.

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
Fold							
0	0.8182	0.0000	0.8182	0.8788	0.8061	0.7250	0.7642
1	0.9091	0.0000	0.9091	0.9273	0.9076	0.8625	0.8735
2	0.9091	0.0000	0.9091	0.9273	0.9076	0.8625	0.8735
3	0.7273	0.0000	0.7273	0.8442	0.6826	0.5875	0.6674
4	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	0.9000	0.0000	0.9000	0.9250	0.8971	0.8485	0.8616
6	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
7	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
8	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
9	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Mean	0.9264	0.0000	0.9264	0.9502	0.9201	0.8886	0.9040
Std	0.0893	0.0000	0.0893	0.0552	0.1011	0.1351	0.1119

PyCaret: AutoML

Paso 5: Optimización de hiperparámetros

Para optimizar los hiperparámetros de un modelo y mejorar su rendimiento, utilizamos `tune_model()`.

```
from pycaret.classification import tune_model
tuned_dt = tune_model(dt, optimize = 'Accuracy')
```

Esto intentará diferentes combinaciones de hiperparámetros para el Árbol de Decisión (dt) y se centrará en optimizar la precisión ('Accuracy').

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
Fold							
0	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	0.9091	0.0000	0.9091	0.9273	0.9076	0.8625	0.8735
2	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	0.7273	0.0000	0.7273	0.8442	0.6826	0.5875	0.6674
4	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	0.9000	0.0000	0.9000	0.9250	0.8971	0.8485	0.8616
6	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
7	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
8	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
9	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Mean	0.9536	0.0000	0.9536	0.9696	0.9487	0.9298	0.9402
Std	0.0843	0.0000	0.0843	0.0510	0.0967	0.1276	0.1049

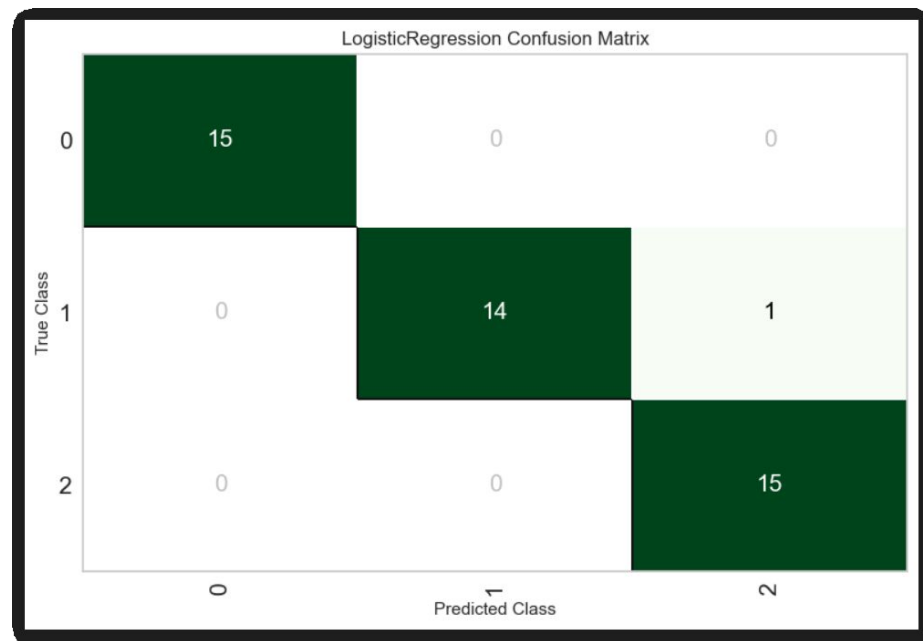
PyCaret: AutoML

Paso 6: Visualización del modelo

PyCaret facilita la visualización de diferentes aspectos de un modelo entrenado. Por ejemplo, para visualizar la matriz de confusión del modelo optimizado:

```
from pycaret.classification import plot_model
plot_model(tuned_dt, plot = 'confusion_matrix')
```

PyCaret soporta una variedad de tipos de gráficas que pueden ser especificadas con el parámetro plot. Algunas opciones incluyen 'auc' para la curva ROC, 'feature' para la importancia de las características, entre otras.



PyCaret: AutoML

Paso 7: Evaluación exhaustiva del modelo

Para una evaluación más detallada que incluya varias métricas y gráficas, PyCaret ofrece `evaluate_model`.

```
from pycaret.classification import plot_model
plot_model(tuned_dt, plot = 'confusion_matrix')
```

Esto abrirá una interfaz de usuario en tu navegador o Jupyter notebook (dependiendo de tu entorno de trabajo) que te permitirá explorar diferentes aspectos del modelo seleccionado.

Plot Type:

Pipeline Plot	Hyperparameters	AUC	Confusion Matrix	Threshold	Precision Recall	Prediction Error
Class Report	Feature Selection	Learning Curve	Manifold Learning	Calibration Curve	Validation Curve	Dimensions
Feature Importance	Feature Importance...	Decision Boundary	Lift Chart	Gain Chart	Decision Tree	KS Statistic Plot



PyCaret: AutoML

Paso 8: Finalizar el modelo

Una vez que estés satisfecho con el rendimiento del modelo, puedes "finalizarlo". Esto entrena el modelo en el conjunto completo de datos (incluyendo el conjunto de prueba).

```
from pycaret.classification import finalize_model  
final_dt = finalize_model(tuned_dt)
```

El modelo finalizado final_dt está listo para ser usado en producción o para hacer predicciones sobre nuevos datos.

Paso 9: Realizar predicciones

Con el modelo finalizado, puedes hacer predicciones sobre nuevos datos. Para demostración, vamos a predecir las etiquetas del mismo conjunto de datos de Iris:

```
from pycaret.classification import predict_model  
predictions = predict_model(final_dt, data=dataset)
```

“predictions” incluirá las etiquetas predichas junto con los datos originales.

PyCaret: AutoML

Paso 10: Guardar y cargar modelos

Finalmente, puedes guardar tu modelo para uso futuro y luego cargarlo cuando sea necesario.

Guardar modelo

```
from pycaret.classification import save_model  
save_model(final_dt, 'final_dt_model_iris')
```

Cargar modelo

```
from pycaret.classification import load_model  
loaded_model = load_model('final_dt_model_iris')
```

El modelo guardado final_dt_model_iris puede ser cargado en cualquier momento para hacer predicciones sobre nuevos conjuntos de datos sin necesidad de reentrenar el modelo.

Azure: Crear cuenta en Azure

Build in the cloud with an Azure free account

Create, deploy, and manage applications across multiple clouds, on-premises, and at the edge

Start free

Pay as you go



Iniciar sesión

Correo electrónico, teléfono o Skype

¿No tiene una cuenta? Cree una.

¿No puede acceder a su cuenta?



Crear cuenta

jose.dmc.mle@gmail.com

Obtener una nueva dirección



← jose.dmc.mle@gmail.com

Creación de una contraseña

Es necesario escribir la contraseña que se quiera utilizar con la cuenta.

Crea una contraseña

☐ Mostrar contraseña

Siguiente



← jose.dmc.mle@gmail.com

Crear

Si un
de na

País o

Perú

Fecha

Día

Una cu
impon
privac
nuestra
https://



← jose.dmc.mle@gmail.com

Comprob electrónico

Es necesario es
jose.dmc.mle@
electrónico, es p
de correo elect
intentarlo.

Escribe el código

☐ Me gustaría
ofertas de l

Al elegir la opción S
Microsoft y la Decla



← jose.dmc.mle@gmail.com

Crear cuenta

Resuelva el rompecabezas para que sepamos que no es un robot.

92717c194da37d964.5350856802



Siguiente



Azure: Crear cuenta de Azure

Create your Azure free account



Popular services free for 12 months



55+ services always free



USD200 credit to use in your first 30 days

No automatic charges

After your credit is over, we'll ask you if you want to continue with pay-as-you-go. If you do, you'll only pay if you use more than the free amounts of services.

Your profile

Country/Region

Peru

Choose the location that matches your billing address. **You cannot change this selection later.** If your country is not listed, the offer is not available in your region. [Learn More](#)

First name

José

Middle name (Optional)

Last name

Mejía

Email address

jose.dmc.mle@gmail.com

Phone

Example: 912 345 678

Lima

Postal Code (Optional)

I understand that Microsoft may contact me about my free account.

- ☐ I agree to the [customer agreement](#).
- ☐ I would like to receive information, tips, and offers about Azure and other Microsoft products and services.
- ☐ I would like Microsoft to share my information with select partners so I can receive relevant information about their products and services.

[Privacy Statement](#)

Next

Identity verification by card

Sign up

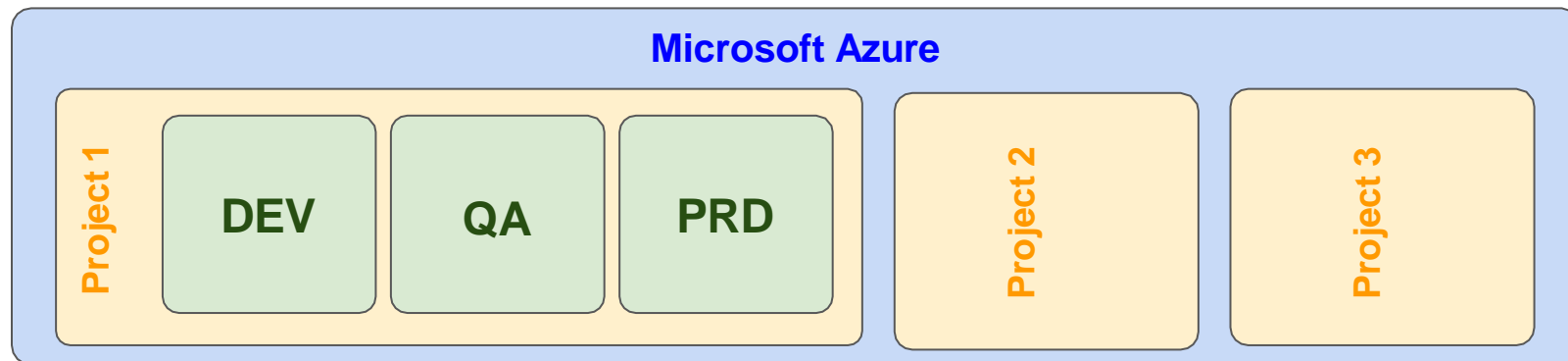
Azure: Conceptos

Subscription

Una suscripción en Azure representa un **acuerdo con Microsoft** para utilizar sus servicios de nube. Esencialmente, actúa como un **contenedor en el que se alojan los recursos** de Azure utilizados. Al crear una suscripción, se **acuerda pagar por los recursos** que consume o por un nivel de servicio específico, según el tipo de suscripción que elija.

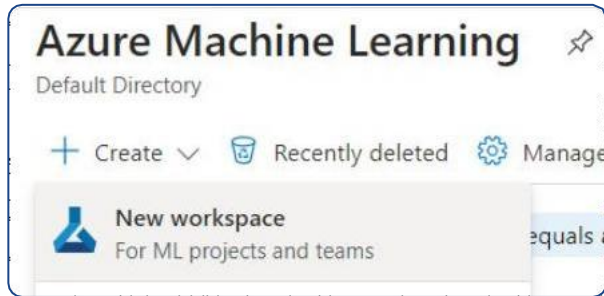
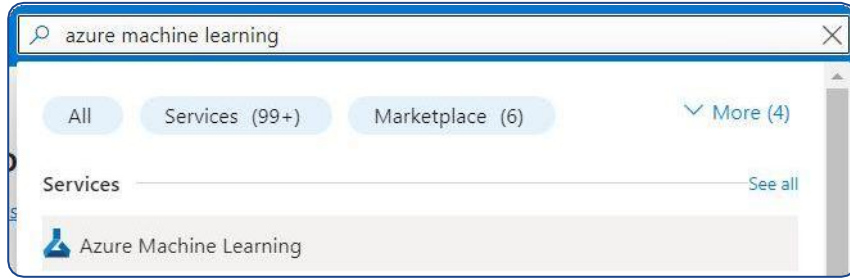
Resource Group

En Azure, un "Resource Group" o Grupo de Recursos es un **contenedor que alberga recursos relacionados para una solución de Azure**. Los recursos son elementos de Azure como aplicaciones web, bases de datos, cuentas de almacenamiento, redes virtuales, etc. Un Grupo de Recursos es una forma de organizar estos recursos en la arquitectura de Azure, facilitando su gestión, monitoreo y acceso al control.





Azure ML: Crear workspace



Resource group * ⓘ

Create new

workspace details

Configure your basic workspace settings like its storage connection, authentication, container, and more. [Learn more](#)

Name * ⓘ

dmc-group ✓

OK Cancel

Subscription * ⓘ

Azure subscription 1

Resource group * ⓘ

Create new

Workspace details

Configure your basic workspace settings like its storage connection, authentication, container, and more. [Learn more](#)

Name * ⓘ

jose-dmc ✓

Region * ⓘ

East US

Storage account * ⓘ

(new) josedmc5143880949

Create new

Key vault * ⓘ

(new) josedmc2333060113

Create new

Application insights * ⓘ

(new) josedmc8235965769

Create new

Container registry ⓘ

None



Azure ML: Crear workspace

Basics Networking Encryption Identity Tags Review + create

Network isolation

Choose the type of network isolation you need for your workspace, from not isolated at all to an entirely separate virtual network managed by Azure Machine Learning. [Learn more about managed network isolation](#)

Public

- Workspace is accessed via public endpoint
- Compute can access public resources
- Outbound data movement is unrestricted

Private with Internet Outbound

- Workspace is accessed via private endpoint
- Compute can access private resources
- Outbound data movement is unrestricted

Private with Approved Outbound

- Workspace is accessed via private endpoint
- Compute can access allowlisted resources only
- Outbound data movement is restricted to approved targets

Basics Networking Encryption Identity Tags Review + create

Data encryption

Azure Machine Learning service stores metrics and metadata in an Azure Cosmos DB instance where all data is encrypted at rest. By default, the data is encrypted with Microsoft-managed keys. You may choose to bring your own (customer-managed) keys.

Encryption type ☐ Microsoft-managed keys ☐ Customer-managed keys

Warning: When using a customer-managed key, the costs for your subscription will be higher because of the additional resources in your subscription. To estimate the cost, use the Azure pricing calculator. To learn more, see [Use customer-managed keys - Azure Machine Learning | Microsoft Docs](#)

Basics Networking Encryption Identity Tags Review + create

Managed identity

A managed identity enables Azure resources to authenticate to cloud services without storing credentials in code. Once enabled, all necessary permissions can be granted via Azure role-based access control. A workspace can be given either a system assigned identity or a user assigned identity.

Identity type ☒ System assigned identity ☐ User assigned identity

Warning: The managed user assigned identity option is only supported if an existing storage account, key vault, and container registry are used.

Storage account access

Azure machine learning allows you to choose between credential-based or identity-based access when connecting to the default storage account. When using identity-based authentication, the Storage Blob Data Contributor role must be granted to the workspace managed identity on the storage account. [Learn more](#)

Storage account access type ☒ Credential-based access ☐ Identity-based access

Data impact

If your workspace contains sensitive data, you can specify a high business impact workspace. This will control the amount of data Microsoft collects for diagnostic purposes and enables additional encryption in Microsoft managed environments.

High business impact workspace ☐

Basics Networking Encryption Identity Tags Review + create




Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups. [Learn more about tags](#)

Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.


Name : Value









Azure ML: Crear workspace


 **Microsoft.MachineLearningServices** | Overview  


Deployment





 Delete  Cancel  Redeploy  Download  Refresh


 Overview

 Inputs


 Outputs











 Template


 **Your deployment is complete**

 Deployment name : Microsoft.MachineLearningServices
Subscription : [Azure subscription 1](#)
Resource group : [dmc-group](#)

Start time : 3/30/2024, 12:05:36 PM
Correlation ID : da55922e-9198-4d12-921d-

 Deployment details

Resource	Type	Status
 jose-dmc	 Azure Machine Learning worksp	OK
 josedmc8235965769	 Application Insights	OK
 josedmc5143880949	 Storage account	OK
 josedmc2333060113	 Key vault	OK
 josedmc8614006025	 Log Analytics workspace	OK

 Next steps

[Go to resource](#)



Azure ML: Launch Studio



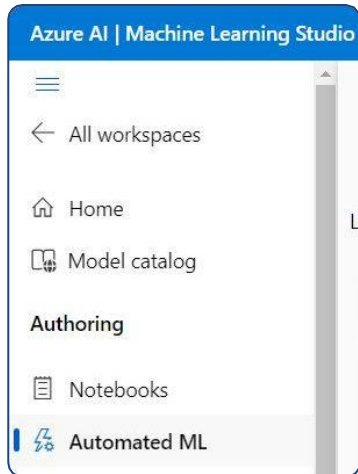
Work with your models in Azure Machine Learning Studio

The Azure Machine Learning Studio is a web app where you can build, train, test, and deploy ML models. Launch it now to start exploring, or [learn more about the Azure Machine Learning studio](#)

[Launch studio](#)

Automated ML

Let Automated ML train and find the best model base

[+ New Automated ML job](#) [Refresh](#)

✓ Training method

1 Basic settings

2 Task type & data

3 Task settings

4 Compute

5 Review

Basic settings

Let's start with some basic information about your training job.

Job name *

jose_demo_xcqlgtp6lj

Experiment name *

Select existing

Create new

New experiment name *

jose_experiment

Description

Tags

Name

:

Value

Add



Azure ML: Launch Studio

Default Directory > jose-dmc > Training job

Submit an Automated ML job PREVIEW

- Training method
- Basic settings
- Task type & data**
- Task settings
- Compute
- Review

Task type & data

Choose the type of task that you would like your model to perform and the data to use for training.

Select task type * ⓘ

Classification

Select data

Make sure your data is preprocessed into a supported format.

[+ Create](#) [Refresh](#) ☒ Show supported data assets only [Reset](#)

Create data asset

- Data type**
- Data source**

Choose a source for your data asset

Choose the data source you want to create your asset from. A data source can be from a local storage location on your computer, from an attached datastore, from Azure publicly available web location.

From Azure storage

Create a data asset from registered data storage services including Azure Blob Storage, Azure file share, and Azure Data Lake.

From local files

Create a data asset by uploading files from your local drive.

From SQL databases

Create a dataset from Azure SQL database and Azure PostgreSQL database.

From web files

Create a data asset from a single file located at a public web URL.

Create data asset

- Data type**
- Data source**

Set the name and type for your data asset

Name *

iris_data

Description

Data asset description

Type * ⓘ

Tabular

Data asset types (from Azure ML v2 APIs)

Table (mitable)

Dataset types (from Azure ML v1 APIs)

Tabular

Create data asset

- Data type**
- Data source**
- Destination storage type**
- MLTable selection**
- Review**

Select a datastore

Choose a storage type and a datastore to upload your data to in the next step. You can create a new datastore or select an existing one.

Datastore type *

Azure Blob Storage ⓘ Create new datastore

Name ↓	Storage name
<input checked="" type="checkbox"/> workspaceblobstore	josedmc5143880949
workspaceartifactstore	josedmc5143880949



Azure ML: Launch Studio

Create data asset

- ✓ Data type
- ✓ Data source
- ✓ Destination storage type
- 4 File or folder selection**
- 5 Settings
- 6 Schema

Choose a file or folder

Choose files or folders to upload from your local drive. If you upload multiple folders or files, they will be stored in a containing folder.

Upload path

azureml://subscriptions/37520400-632e-413e-b571-e52d252eb3c6/resourcegroups/dmc-group/wor...

Upload files or folder ▾

Choose a file or folder

Choose files or folders to upload from your local drive. If you upload multiple folders or files, they will be stored in a containing folder.

Upload path

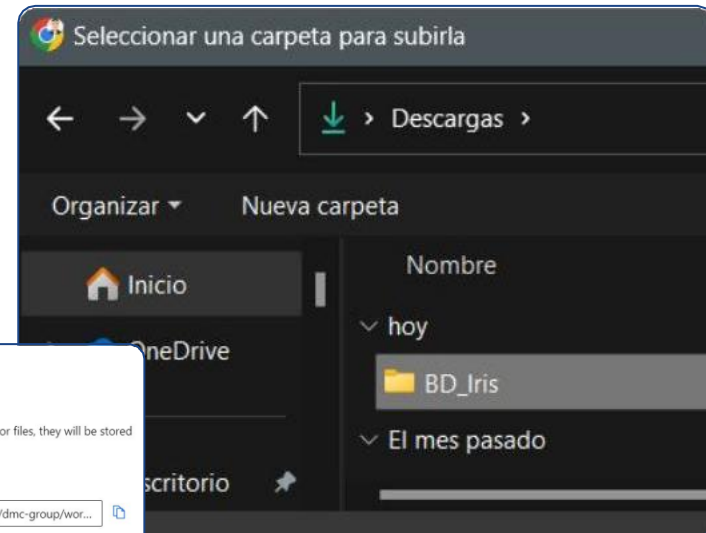
azureml://subscriptions/37520400-632e-413e-b571-e52d252eb3c6/resourcegroups/dmc-group/wor...

Upload files or folder ▾

☐ Overwrite if already exists

Upload list

iris_dataset.csv	2.85 KB/2.85 KB	...
------------------	-----------------	-----





Azure ML: Launch Studio

Create data asset ✕

✓ Data type

✓ Data source

✓ Destination storage type

✓ File or folder selection

5 Settings

6 Schema

7 Review

Settings

These settings determine how the data is parsed. The initial settings are automatically detected; you can change them as needed to reparse the data.

File format

Delimited

Delimiter

Comma

Example

Field1,Field2,Field3

Encoding

UTF-8

Column headers

All files have same headers

Skip rows

None

☒ Dataset contains multi-line data ⓘ

ⓘ Note: Processing tabular files with multi-line data is slower because multiple CPU cores cannot be used to ingest the data in parallel. Checking this option may result in slower processing times.

Data preview

sepal_length_cm	sepal_width_cm	petal_length_cm	petal_width_cm	target
5.1	3.5	1.4	0.2	0
4.9	3	1.4	0.2	0
4.7	3.2	1.3	0.2	0
4.6	3.1	1.5	0.2	0
5	3.6	1.4	0.2	0
5.4	3.9	1.7	0.4	0

Back

Next

Review

Cancel



Azure ML: Launch Studio

Create data asset

✓ Data type

✓ Data source

✓ Destination storage type

✓ File or folder selection

✓ Settings

6 Schema

7 Review

Schema

Column types are auto-detected based on the initial subset of the data and can be updated here. Any conversions preview errors are non-blocking and you

Search column name

Include	Column name	Type	Example values
<input type="checkbox"/>	Path	String	
<input checked="" type="checkbox"/>	sepal_length_cm	Decimal (dot ':')	5.1, 4.9, 4.7
<input checked="" type="checkbox"/>	sepal_width_cm	Decimal (dot ':')	3.5, 3, 3.2
<input checked="" type="checkbox"/>	petal_length_cm	Decimal (dot ':')	1.4, 1.4, 1.3
<input checked="" type="checkbox"/>	petal_width_cm	Decimal (dot ':')	0.2, 0.2, 0.2
<input checked="" type="checkbox"/>	target	Integer	0, 0, 0

Back

Next



Azure ML: Launch Studio

Submit an Automated ML job PREVIEW

✓ Training method

✓ Basic settings

3 Task type & data

4 Task settings

5 Compute

6 Review

Task type & data

✓ Success: iris_data data asset created successfully. It may take a few seconds for the data to be available.

Choose the type of task that you would like your model to perform and the data to use.

Select task type * ⓘ

Select task type ▼ *

Select data

Make sure your data is preprocessed into a supported format.

+ Create

↻ Refresh

☒ Show supported data assets

🔍 Search

Name	Type
✓ iris_data	Table

Submit an Automated ML job PREVIEW

✓ Training method

✓ Basic settings

✓ Task type & data

4 Task settings

5 Compute

6 Review

iris_data [\(View data\)](#)

Target column *

target (Integer)

Classification settings

☐ Enable deep learning ⓘ

⚙️ View additional configuration settings

📄 View featurization settings

> Limits

Validate and test

You can choose a validation type and select test data as an optional step.

Validation type ⓘ

Automatic ▼

Test data ⓘ

None ▼

Back

Next



Azure ML: Launch Studio

Submit an Automated ML job PREVIEW

✓ Training method

✓ Basic settings

✓ Task type & data

✓ Task settings

5 Compute

6 Review

Compute

Select and configure the compute resource for executing your training job.

Select compute type

Serverless

Virtual machine type ⓘ

☒ CPU ☐ GPU

Virtual machine tier ⓘ

☒ Dedicated ☐ Low priority

Virtual machine size

Standard_D2_v3 (2 core(s), 8GB RAM, 50GB storage, \$0.10/hr)

Number of instances

1



Azure ML: Launch Studio

Default Directory > jose-dmc > Jobs > jose_experiment > jose_demo_6d74bfp2xj

jose_demo_6d74bfp2xj Running

Overview Data guardrails Models + child jobs Outputs + logs Child jobs

Refresh Edit and submit (preview) Register model Cancel Delete | Compare (preview)

Properties

Status Running <div>Setting up the run</div>	Job type Automated ML
Created on Mar 30, 2024 6:52 PM	Experiment jose_experiment
Start time Mar 30, 2024 6:52 PM	Arguments None
Name jose_demo_6d74bfp2xj	See all properties Raw JSON
Script name --	See YAML job definition Job YAML
Created by José Mejía Gamarra	

Inputs

Input name: training_data
Data asset: [iris_data:1](#)
Asset URI:

Best model summary

No data

Run summary

Task type
Classification [View configuration settings](#)

Featurization
Auto

Primary metric
AUC weighted



Azure ML: Launch Studio

Overview

Data guardrails

Models + child jobs

Outputs + logs

Child jobs

Refresh

Edit and submit (preview)

+ Register model

Cancel

Delete

Compare (preview)

Properties

Status

Completed

Warning: No scores improved over last 20 iterations, so experiment stopped early. This early stopping behavior can be disabled by

See more details

Created on

Mar 30, 2024 6:58 PM

Start time

Mar 30, 2024 6:59 PM

Duration

45m 24.60s

Compute duration

45m 24.60s

Name

Created by

José Mejía Gamarra

Job type

Automated ML

Experiment

jose_experiment

Arguments

None

See all properties

Raw JSON

See YAML job definition

Job YAML

Inputs

Input name: training_data

Data asset: iris_data:1

Asset URI: azureml:iris_data:1

Outputs

Output name: best_model

Model: azureml_strong_kiwi_rc3jcbd9wt_2_output_mlflow_log_model_835912939:1

Asset URI: azureml:azureml_strong_kiwi_rc3jcbd9wt_2_output_mlflow_log_model_8359...

Output name: full_training_dataset

Dataset: 7fbcc207-f97f-4271-a002-7cf77e6cf117

Best model summary

Algorithm name

MaxAbsScaler, ExtremeRandomTrees



Azure ML: Launch Studio

Best model summary

Algorithm name

MaxAbsScaler, ExtremeRandomTrees

Hyperparameters

[View hyperparameters](#)

AUC weighted

0.99856 [View all other metrics](#)

Sampling

100.00 % ⓘ

Registered models

No registration yet

Deploy status

No deployment yet

Overview

Model

Explanations (preview)

Responsible AI (preview)

Metrics

Data transformation (preview)

Test results (preview)

Outputs + logs

Images

Child jobs

...

↻ Refresh

▶ Deploy ▾

⬇ Download

🔍 Explain model

View generated code

✓ Test model (preview)

+ Register model

⊗ Cancel

🗑 Delete

Model summary

Algorithm name

MaxAbsScaler, ExtremeRandomTrees

Hyperparameters

[View hyperparameters](#)

AUC weighted

0.99856 [View all other metrics](#)

Sampling

100.00 % ⓘ

Registered models

No registration yet

Deploy status

No deployment yet



Azure: Instalar CLI

<https://learn.microsoft.com/en-us/cli/azure/install-azure-cli>

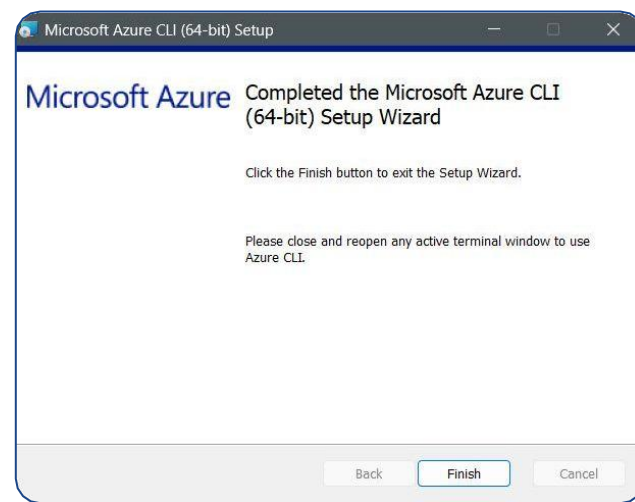
Latest version

Download and install the latest release of the Azure CLI. When the installer asks if it can make changes to your computer, select the "Yes" box.

Latest MSI of the Azure CLI (32-bit)

Latest MSI of the Azure CLI (64-bit)

If you have previously installed the Azure CLI, run `az --help` to overwrite an existing installation.





Azure: AutoML with Python SDK

Step 0: Install dependencies

```
# !pip install azure-identity==1.15.0
# !pip install azureml-fsspec==1.3.1
# !pip install azureml-sdk==1.55.0
# !pip install azure-ai-ml==1.15.0
# !pip install mltable==1.6.1
```

Step 1: Set up your workspace

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

credential = DefaultAzureCredential()
ml_client = None
try:
    # Enter details of your Azure Machine Learning workspace
    subscription_id = "subscription-id-413e-b571-e52d252eb3c6"
    resource_group = "dmc-group"
    workspace = "jose-dmc"
    ml_client = MLClient(credential, subscription_id, resource_group, workspace)
except Exception as ex:
    print(ex)
```

Step 2: Data source and format

```
import mltable

paths = [
    {'file': './train_data/iris_dataset.csv'}
]

train_table = mltable.from_delimited_files(paths)
train_table.save('./train_data')
```

✓ 1.9s Python

```
paths:
- file: file:///d:/DMC/AutoML_Azure/train_data/iris_dataset.csv
transformations:
- read_delimited:
  delimiter: ','
  empty_as_string: false
  encoding: utf8
  header: all_files_same_headers
  include_path_column: false
  infer_column_types: true
  partition_size: 20971520
  path_column: Path
  support_multi_line: false
type: mltable
```




Azure: AutoML with Python SDK

Step 3: Create a Compute Instance (If not exist)

```
# #Sign into Azure with Azure CLI  
# !az login
```

✓ 0.0s

Python

```
ci_basic_name = "jose-basic-ci"  
from azure.ai.ml.entities import ComputeInstance, AmlCompute  
ci_basic = ComputeInstance(name=ci_basic_name, size="Standard_DS2_v2")  
ml_client.begin_create_or_update(ci_basic).result()
```

✓ 1m 37.5s

Python

```
ComputeInstance({'state': 'Running', 'last_operation': {'operation_name': 'Create', '}}
```




Azure: AutoML with Python SDK

Step 4: Configure your experiment settings

```
from azure.ai.ml.constants import AssetTypes
from azure.ai.ml import automl, Input

# make an Input object for the training data
my_training_data_input = Input(
    type=AssetTypes.MLTABLE, path="./train_data"
)

# configure the classification job
classification_job = automl.classification(
    compute           = ci_basic_name,
    experiment_name    = "jose_experiment_python_3",
    training_data      = my_training_data_input,
    target_column_name = "target",
    primary_metric     = "accuracy",
    n_cross_validations = 5,
    enable_model_explainability = True,
    tags               = {"dmc": "demo"}
)

# # Limits are all optional
# classification_job.set_limits(
#     timeout_minutes=600,
#     trial_timeout_minutes=20,
#     max_trials=2,
#     enable_early_termination=True,
# )

# # Training properties are optional
# classification_job.set_training(
#     blocked_training_algorithms=["logistic_regression"],
#     enable_onnx_compatible_models=True
# )
```



Azure: AutoML with Python SDK

Step 5: Run experiment

```
# Submit the AutoML job
returned_job = ml_client.jobs.create_or_update(classification_job)

# Get the URL to monitor the job in Azure Machine Learning studio
print(f"Monitor your job at {returned_job.studio_url}")
```

✓ 10.2s

Monitor your job at https://ml.azure.com/runs/quirky_star_nktb75xkbp?wsid=



Azure: AutoML with Python SDK

quirky_star_nktb75xkkp Completed

Overview | Data guardrails | Models + child jobs | Outputs + logs | Child jobs

Refresh | Edit and submit (preview) | Register model | Cancel | Delete | Compare (preview) ▾

Properties

Status Completed ▾ Warning: No scores improved over last 20 iterations, so experiment stopped early. This early stopping behavior can be disabled by See more details	Job type Automated ML Experiment jose_experiment_python_3
Created on Mar 31, 2024 8:51 AM	Arguments None
Start time Mar 31, 2024 8:51 AM	Git repository https://github.com/EnriqueMejia96/AutoML_Azure.git
Duration 30m 43.39s	Git branch main
Compute duration 30m 43.39s	Git commit 3023e7f374fc3fad9f784dd344cf47af645d9644
Compute target	See all properties

Inputs

Input name: training_data
Data asset: azureml_quirky_star_nktb75xkkp_input_data_TrainData:1
Asset URI:

Outputs

Output name: best_model
Model: azureml_quirky_star_nktb75xkkp_40_output_mlflow_log_model_835912939:1
Asset URI:

Output name: full_training_dataset
Dataset: b157c272-c6d4-4912-85a0-1e40c05233b6

Best model summary

Algorithm name
[VotingEnsemble](#)