

Tarea evaluable

CE_5075 8.1

Apartado 3

Big data aplicado



Índice

APARTADO 3	2
1 - CONFIGURACIÓN DEL CLÚSTER	2
2 - CREAR UN EXPERIMENTO	4
3 - REGISTRAR UN MODELO	6

APARTADO 3

Vamos a trabajar con el dataset de pingüinos del archipiélago Palmer, que se puede descargar desde Kaggle (archivo penguins_size.csv), con el cual ya hemos trabajado en el módulo de Programación de IA. También puedes descargar el archivo desde:

https://raw.githubusercontent.com/tnavarrete-iedib/bigdata-24-25/refs/heads/main/penguins_size.csv.

Este dataset tiene 6 columnas de entrada, en este orden:

- island: nombre de la isla (Dream, Torgersen o Biscoe)
- culmen_length_mm: longitud del pico en mm
- culmen_depth_mm: profundidad del pico en mm
- flipper_length_mm: longitud de la aleta en mm
- body_mass_g: masa corporal en gramos
- sex: sexo (MALE o FEMALE)

Fíjate que en los datos hay valores ausentes, etiquetados como NA. Elimina toda la fila si tiene un NA en cualquiera de sus columnas.

También hay una columna llamada species, que contiene la etiqueta de clase correspondiente, y que representa una de estas tres especies de pingüinos:

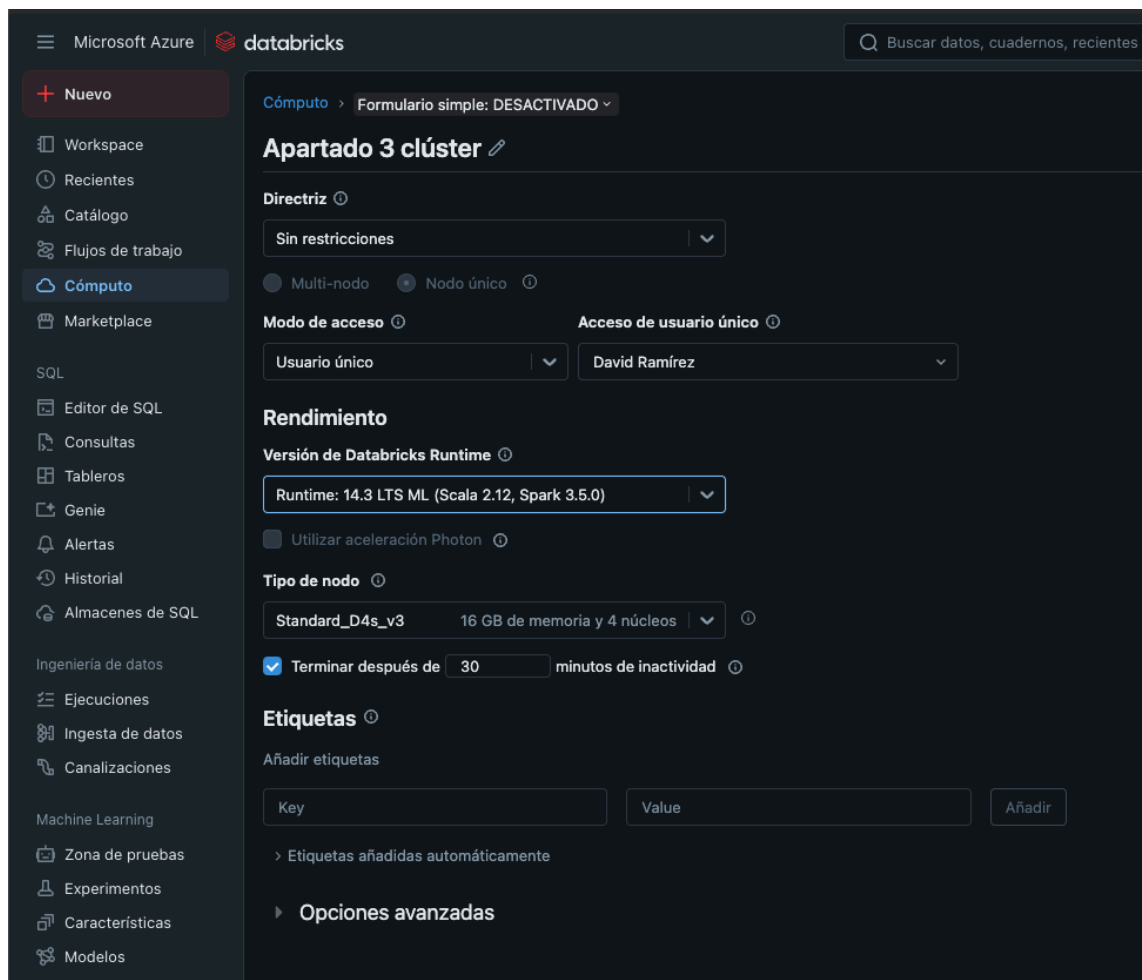
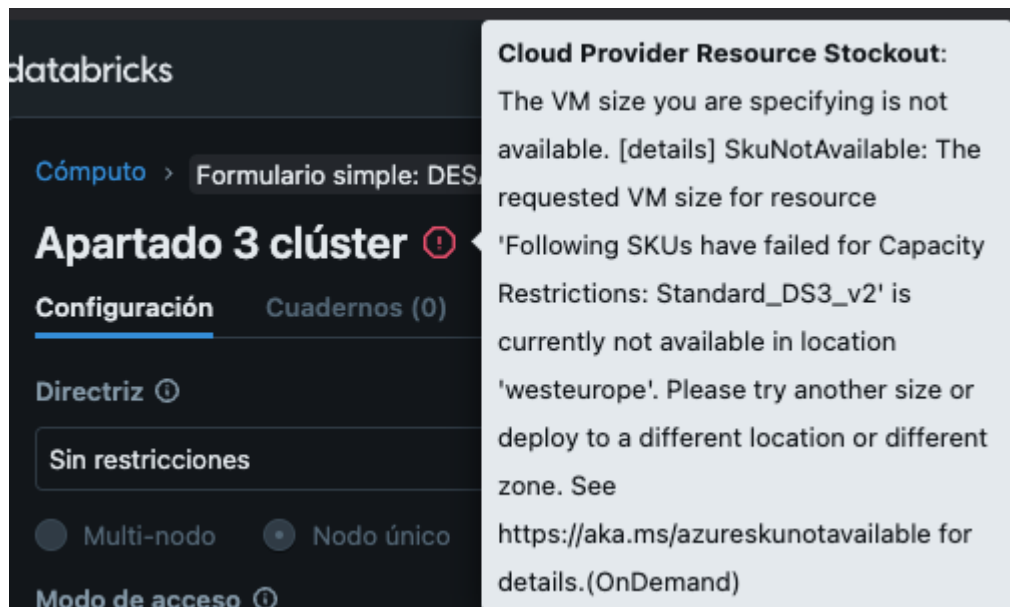
- Chinstrap
- Adélie
- Gentoo

Se nos pide usar AutoML en Azure Databricks para obtener un modelo de clasificación (la variable objetivo es la especie del pingüino). Utiliza la precisión (accuracy) como métrica. Puedes limitar la duración del experimento a 10 minutos. Finalmente, genera un endpoint y Pruébalo, enviando una petición con los datos de varios pingüinos.

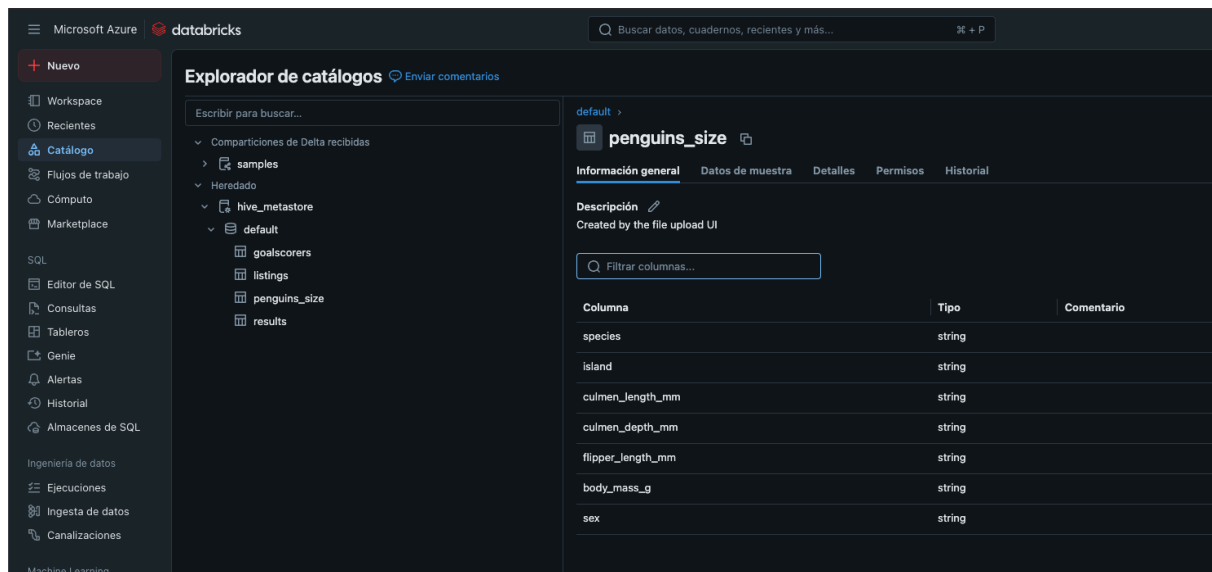
1 - CONFIGURACIÓN DEL CLÚSTER

Al igual que en los apartados anteriores, primero es necesario configurar el entorno. Vamos a crear un nuevo clúster siguiendo los parámetros especificados en los apuntes.

He vuelto a tener problemas al intentar usar "Standard_DS3_v2" por lo que al igual que en el apartado 1 usaré "Standard_D4s_v3"

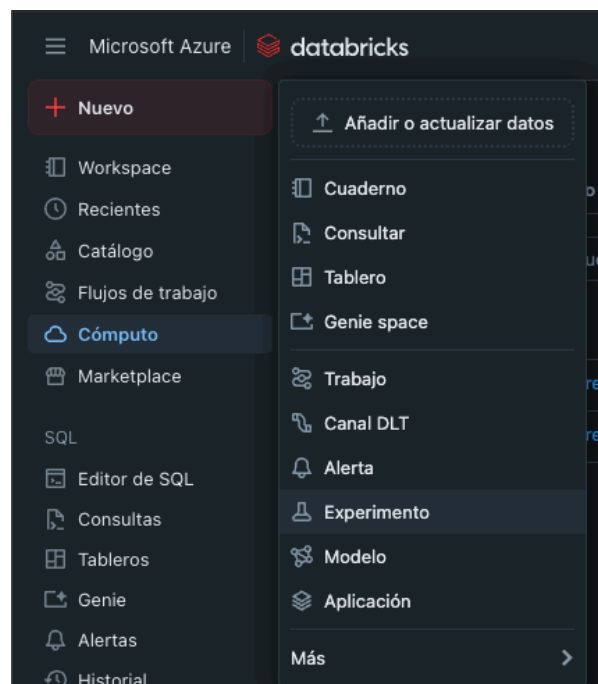


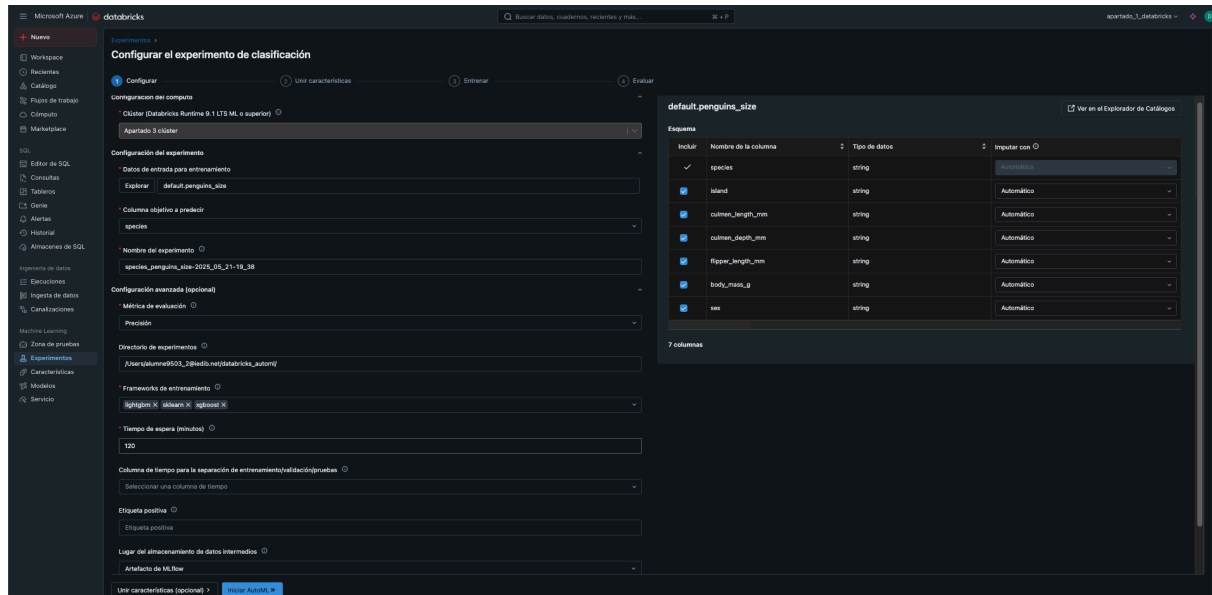
Después de crear el clúster cargamos los datos de penguins_size.csv



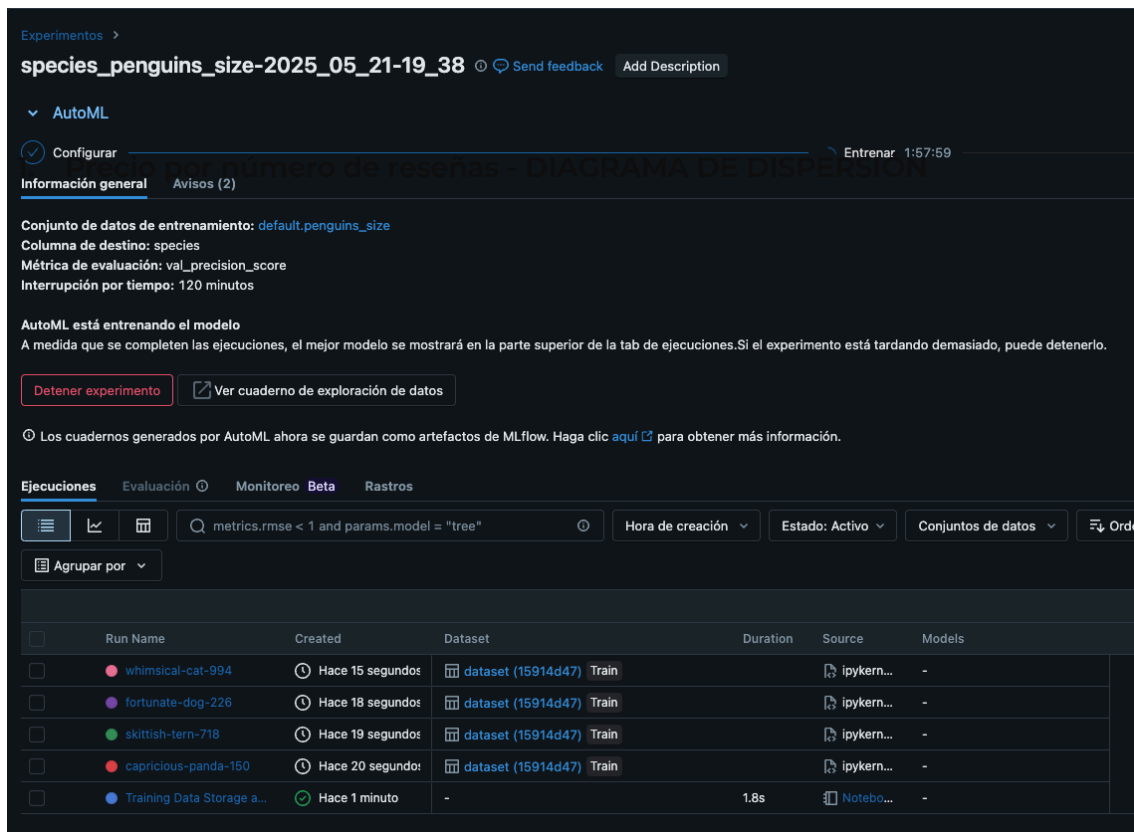
2 - CREAR UN EXPERIMENTO

Seleccionamos “Nuevo” y “Experimento” en el menú lateral. En las siguientes imágenes se mostrarán los pasos a seguir para crear un experimento de tipo clasificación.





Después de iniciar Auto ML veremos en pantalla una barra de progreso con el tiempo restante del entrenamiento además de ver cómo se registran varios experimentos



Experimentos > **species_penguins_size-2025_05_21-19_38** Send feedback [Add Description](#)

▼ AutoML

✓ Configurar Entrenar 1:27:51

Información general Avisos (2)

Conjunto de datos de entrenamiento: default.penguins_size
 Columna de destino: species
 Métrica de evaluación: val_precision_score
 Interrupción por tiempo: 120 minutos

AutoML está entrenando el modelo
 A medida que se completan las ejecuciones, el mejor modelo se mostrará en la parte superior de la tab de ejecuciones. Si el experimento está tardando demasiado, puede detenerlo.

[Detener experimento](#) [Ver cuaderno de exploración de datos](#)

🔗 Los cuadernos generados por AutoML ahora se guardan como artefactos de MLflow. Haga clic [aquí](#) para obtener más información.

Ejecuciones Evaluación Monitoreo Beta Rastros

Hora de creación Estado: Activo Conjuntos de datos Ordenar: val_precision_score

☐ Expandir filas ☐ Agrupar por ▼

	Run Name	Created	Dataset	Duration	Source	Models	Metrics	Tags
							val_precisi	model_type
<input type="checkbox"/>	clumsy-ant-466	✓ Hace 2 minutos	dataset (15914d47) Train ds... +2	1.1min	-	sklearn	1	logistic_reg...
<input type="checkbox"/>	nebulous-fly-393	✓ Hace 3 minutos	dataset (15914d47) Train ds... +2	1.1min	-	sklearn	1	logistic_reg...
<input type="checkbox"/>	debonair-asp-847	✓ Hace 4 minutos	dataset (15914d47) Train ds... +2	1.1min	-	sklearn	1	logistic_reg...
<input type="checkbox"/>	omniscient-fly-409	✓ Hace 4 minutos	dataset (15914d47) Train ds... +2	1.1min	-	sklearn	1	logistic_reg...
<input type="checkbox"/>	honorable-midge-948	✓ Hace 5 minutos	dataset (15914d47) Train ds... +2	1.1min	-	sklearn	1	logistic_reg...
<input type="checkbox"/>	legendary-asp-93	✓ Hace 6 minutos	dataset (15914d47) Train ds... +2	1.1min	-	sklearn	1	logistic_reg...
<input type="checkbox"/>	powerful-flea-548	✓ Hace 6 minutos	dataset (15914d47) Train ds... +2	1.1min	-	sklearn	1	logistic_reg...

3 - REGISTRAR UN MODELO

Para registrar un modelo seleccionamos el último modelo generado y pulsamos en el botón arriba a la derecha “Registrar modelo”. En mi caso el experimento ha estado entrenando durante aproximadamente 30 minutos.

Experimentos > /Users/alumne9503_2@iedib.net/databricks_automl/species_penguins_size-2025_05_21-19_38 > **clumsy-ant-466** Send feedback

Información general Métricas del modelo Métricas del sistema Rastros Resultados de la evaluación Artefactos

Descripción

No hay descripción

Detalles

Creación	21 may 2025, 20:14
Creador	alumne9503_2@iedib.net
ID experimento	1059039176075809
Estado	✓ Terminado
ID ejecución	65a9a279f04d4425b7f2f2f698284fe6
Duración	1.1min
Conjuntos de datos utilizados	dataset (15914d47) Train +3
Etiquetas	estimator_class: sklearn.pipeline.Pipeline estimator_name: Pipeline model_type: logistic_regression_classifier
Procedencia	—
Modelos registrados	sklearn
Modelos registrados	—

Métricas (24)

Métricas de búsqueda

Métrica	Más reciente	Min.	Máx.
training_precision_score	1	1	1
training_roc_auc	1	1	1
val_accuracy_score	1	1	1
val_f1_score	1	1	1
val_precision_score	1	1	1
val_roc_auc	1	1	1
test_accuracy_score	0.9859154929577465	0.9859154929577465	0.9859154929577465
test_f1_score	0.9857747182746655	0.9857747182746655	0.9857747182746655
test_precision_score	0.986317907444668	0.986317907444668	0.986317907444668
test_roc_auc	0.9996193376475065	0.9996193376475065	0.9996193376475065
training_accuracy_score	1	1	1
training_f1_score	1	1	1
training_log_loss	0.004183963068115075	0.004183963068115075	0.004183963068115075
training_recall_score	1	1	1
training_score	1	1	1
val_example_count	71	71	71

Parámetros (75)

Parámetros de búsqueda

Parámetro	Valor
classifier	LogisticRegression(C=11.242928749788101, multi_class='multinomial', penalty='l1', random_state=692229712, solver='saga')
classifier__C	11.242928749788101
classifier__class_weight	None
classifier__dual	False
classifier__fit_intercept	True
classifier__intercept_scaling	1
classifier__l1_ratio	None
classifier__max_iter	100
classifier__multi_class	multinomial
classifier__n_jobs	None
classifier__penalty	l1
classifier__random_state	692229712
classifier__solver	saga
classifier__tol	0.0001
classifier__verbose	0
classifier__warm_start	False

Registrar modelo

Model

+ Create New Model

Model Name

penguins-model

Cancelar

Registrar

Microsoft Azure databricks

Buscar datos, cuadernos, recientes y más...

Nuevo

Workspace

Recientes

Catálogo

Flujos de trabajo

Cómputo

Marketplace

SQL

Editor de SQL

Consultas

Tableros

Genie

Alertas

Historial

Almacenes de SQL

Ingeniería de datos

Ejecuciones

Ingesta de datos

Canalizaciones

Machine Learning

Zona de pruebas

Experimentos

Características

Modelos

Servicio

Modelos registrados

penguins-model

Detalles

Notificarme sobre

Toda la actividad nueva

Hora de creación: 21 may 2025, 20:19

Último cambio: 21 may 2025, 20:19

Creador: alumne9503_2@iedib.net

Descripción

Editar

Etiquetas

Versiones

Todo

Activo 0

Comparar

Versión	Registrado en	Creador	Estadio	Solicitudes pendientes	Descripción
<div>✓</div> <div>Versión 1</div>	21 may 2025, 20:19	alumne9503_2@iedib.net	None	-	

Microsoft Azure databricks

Buscar datos, cuadernos, recientes y más...

Nuevo

- Workspace
- Recientes
- Catálogo
- Flujos de trabajo
- Cómputo
- Marketplace

SQL

- Editor de SQL
- Consultas
- Tableros
- Genie
- Alertas
- Historial
- Almacenes de SQL

Ingeniería de datos

- Ejecuciones
- Ingesta de datos
- Canalizaciones

Machine Learning

- Zona de pruebas
- Experimentos
- Características
- Modelos**
- Servicio

Modelos registrados > penguins-model >

Versión 1

Registrado en: 21 may 2025, 20:19 Creador: alumne9503_2@iedib.net Estatus del seguimiento: ▲ Siguiéndose

Estado: **None** Descripción Editar

Solicitudes pendientes

Solicitud	Solicitud realizada por	Acciones
No hay ninguna solicitud pendiente.		

Etiquetas

Esquema

Nombre	Tipo
Entradas (6)	
island (required)	string
culmen_length_mm (required)	string
culmen_depth_mm (required)	string
flipper_length_mm (required)	string
body_mass_g (required)	string
sex (required)	string
Salidas (1)	

Ahora creamos un nuevo endpoint, así es como podremos realizar las predicciones de las especies de pingüinos. A ese endpoint le vamos a asignar este modelo que hemos registrado.

nueva

Comparar

Registrado en 21 may 2025, 20:19

Configurar la inferencia del modelo

Send feedback

En tiempo real Transmisión (Delta Live Tables) Inferencia Batch

Servir este modelo a través de un punto. Para configurar varios modelos servidos desde este punto, actualice su configuración a través de la página de detalles del punto. Más información

Versión del modelo: Version 1

Nombre del punto: penguin

Cómputo: Small 4 concurrency (4 DBU) Escalar a cero

Cancelar Crear endpoint

Microsoft Azure | databricks

Buscar datos, cuadernos, recientes y más...

apartado_1_databricks

Nuevo

- Workspace
- Recientes
- Catálogo
- Flujos de trabajo
- Cómputo
- Marketplace

SQL

- Editor de SQL
- Consultas
- Tableros
- Genie
- Alertas
- Historial
- Almacenes de SQL

Ingeniería de datos

- Ejecuciones
- Ingesta de datos
- Canalizaciones

Machine Learning

- Zona de pruebas
- Experimentos
- Características
- Modelos
- Servicio**

Puntos de servicio

penguin

Estado del punto de servicio: Listo

Creador: David Ramirez

URL: <https://adb-1942207080158662.2.azure.databricks.net/serving-endpoints/penguin/invocations>

Etiquetas:

Política de presupuestos serverless:

Optimización de rutas: Desactivado

Permisos | Editar | Detener | **Usar**

Puerta de enlace Preview

Límites de velocidad: No está configurado

Configurar la puerta de enlace de IA

Configuración activa

Entidad	Versión	Nombre	Estado	Cómputo	Tráfico, %
penguins-model	Versión 1	penguins-model-1	Listo	CPU, Small 4 concurrency (4 DBU)	100

Métricas | **Eventos** | Logs

Timestamp	Tipo de evento	Nombre de la entidad servida	Mensaje
21 may 2025, 20:41	ENDPOINT_EVENT		Endpoint 'penguin' entered READY state.
21 may 2025, 20:41	ENDPOINT_UPDATE_EVENT		Endpoint update succeeded for endpoint 'penguin', config version 1.
21 may 2025, 20:41	SERVED_ENTITY_SERVICE_EVENT	penguins-model-1	Served entity creation succeeded for served entity 'penguins-model-1', config version 1.
21 may 2025, 20:41	SERVED_ENTITY_SERVICE_EVENT		Served entity 'penguins-model-1' entered DEPLOYMENT_READY state.
21 may 2025, 20:38	SERVED_ENTITY_SERVICE_EVENT		Served entity 'penguins-model-1' entered DEPLOYMENT_CREATING state: Deploying

Una vez el endpoint está preparado podemos pulsar “Usar” arriba a la derecha para enviar las peticiones.

Consultar punto de servicio

Navegador | Curl | Python | SQL

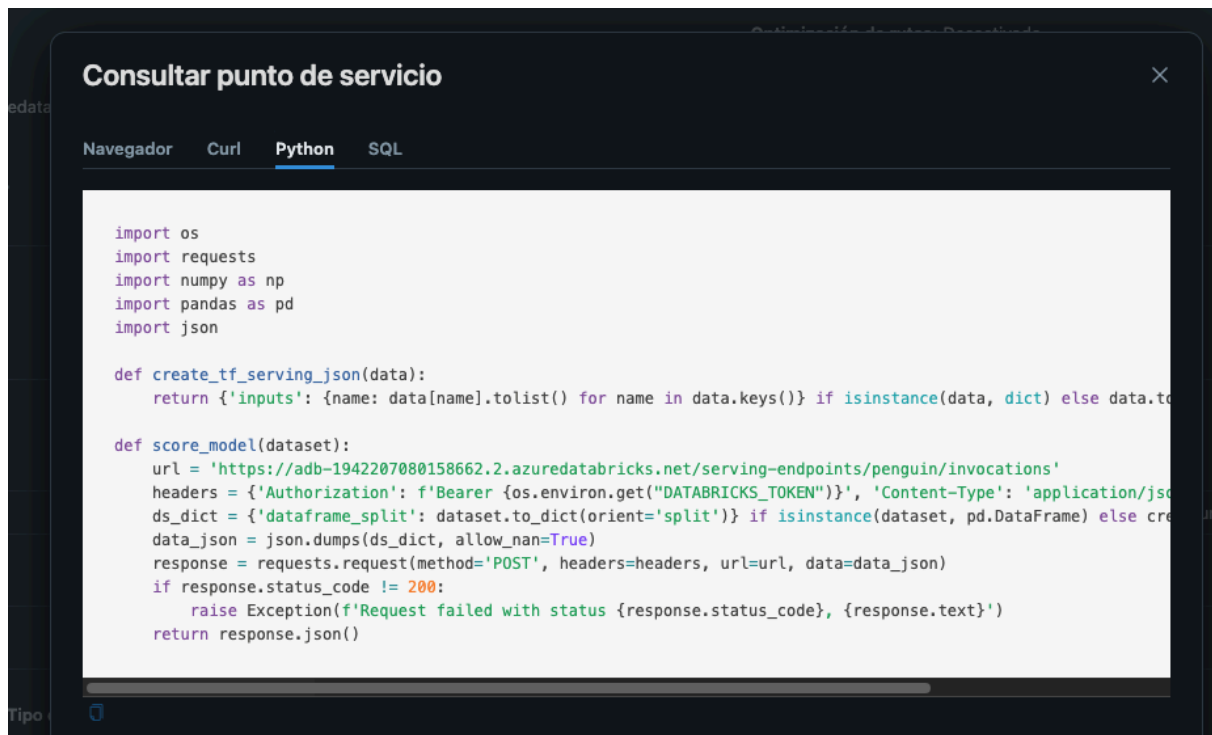
Solicitud

```
{
  "dataframe_split": {
    "columns": [
      "island",
      "culmen_length_mm",
      "culmen_depth_mm",
      "flipper_length_mm",
      "body_mass_g",
      "sex"
    ],
  },
  "data": [
    [
      "Torgersen",
      "39.1",
      "18.7",
      "181",
      "3750",
      "MALE"
    ]
  ]
}
```

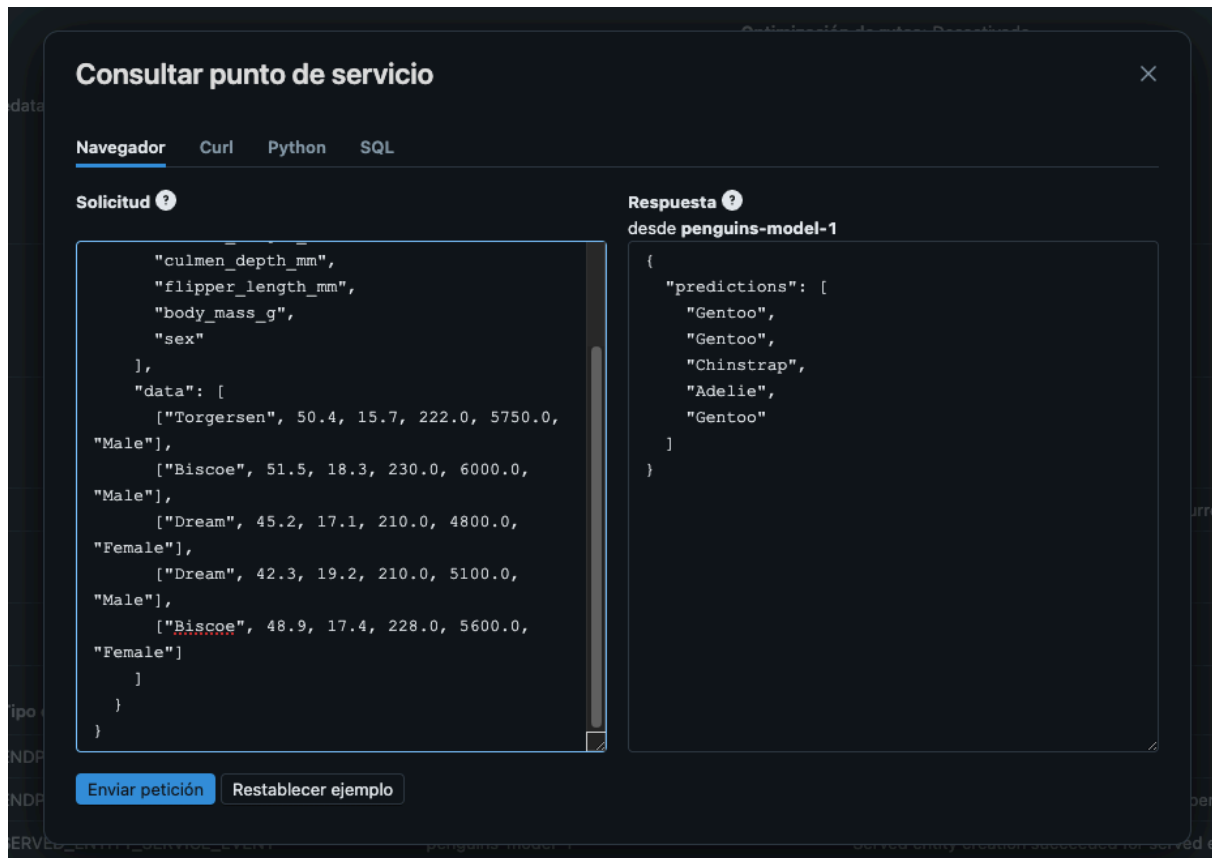
Respuesta desde penguins-model-1

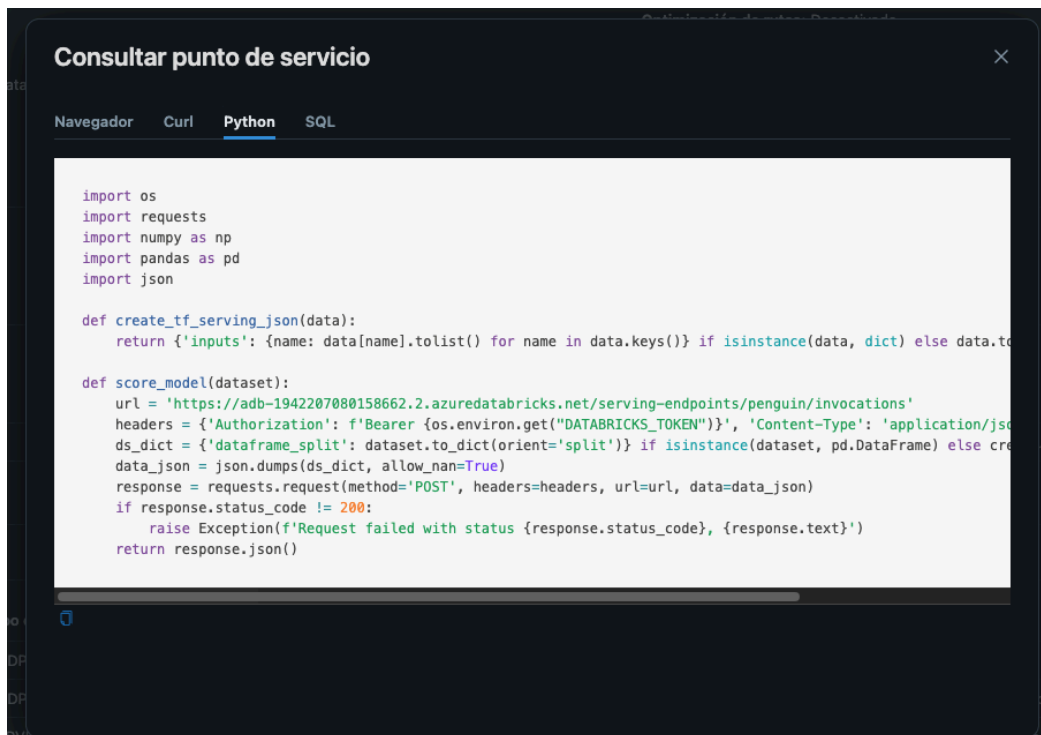
```
{
  "predictions": [
    "Adelie",
    "Adelie",
    "Adelie",
    "Adelie",
    "Adelie"
  ]
}
```

Enviar petición | Restablecer ejemplo



Vemos que nos devuelve la misma especie varias veces. Para comprobar si el modelo funciona correctamente vamos a usar los datos usados en la práctica 4 de Programación de IA.





```
Consultar punto de servicio

Navegador  Curl  Python  SQL

import os
import requests
import numpy as np
import pandas as pd
import json

def create_tf_serving_json(data):
    return {'inputs': {name: data[name].tolist() for name in data.keys() if isinstance(data, dict) else data.to_dict(orient='split')}}

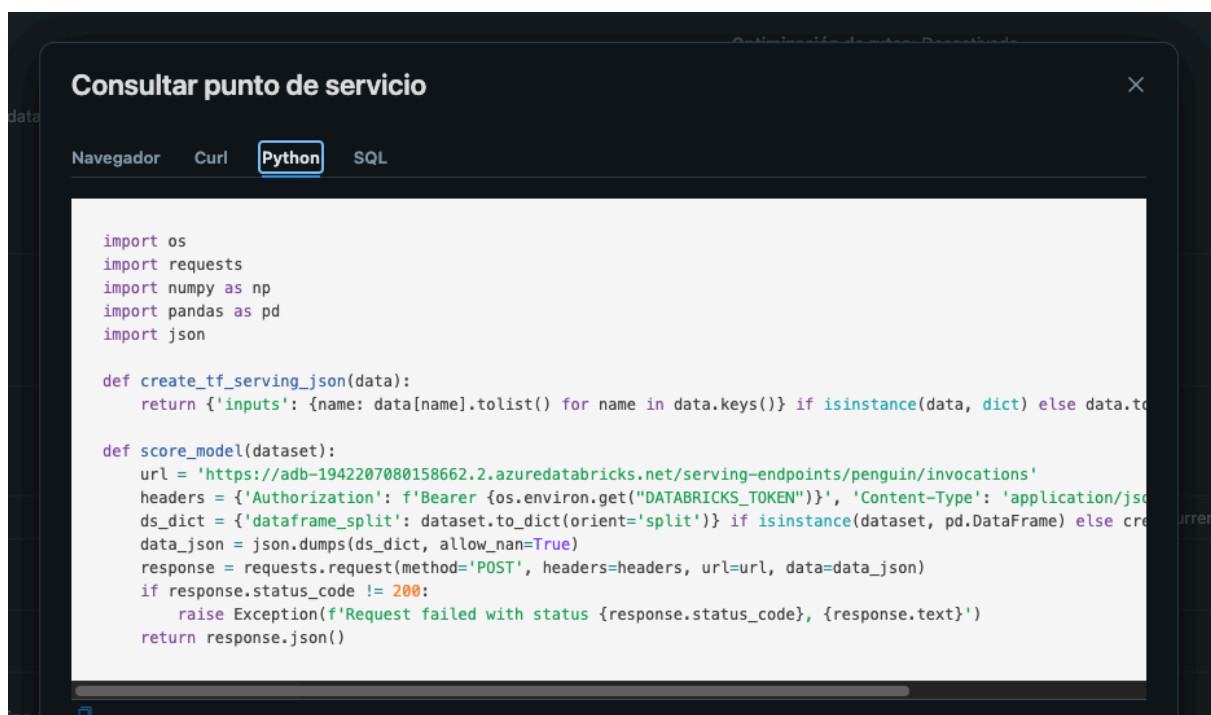
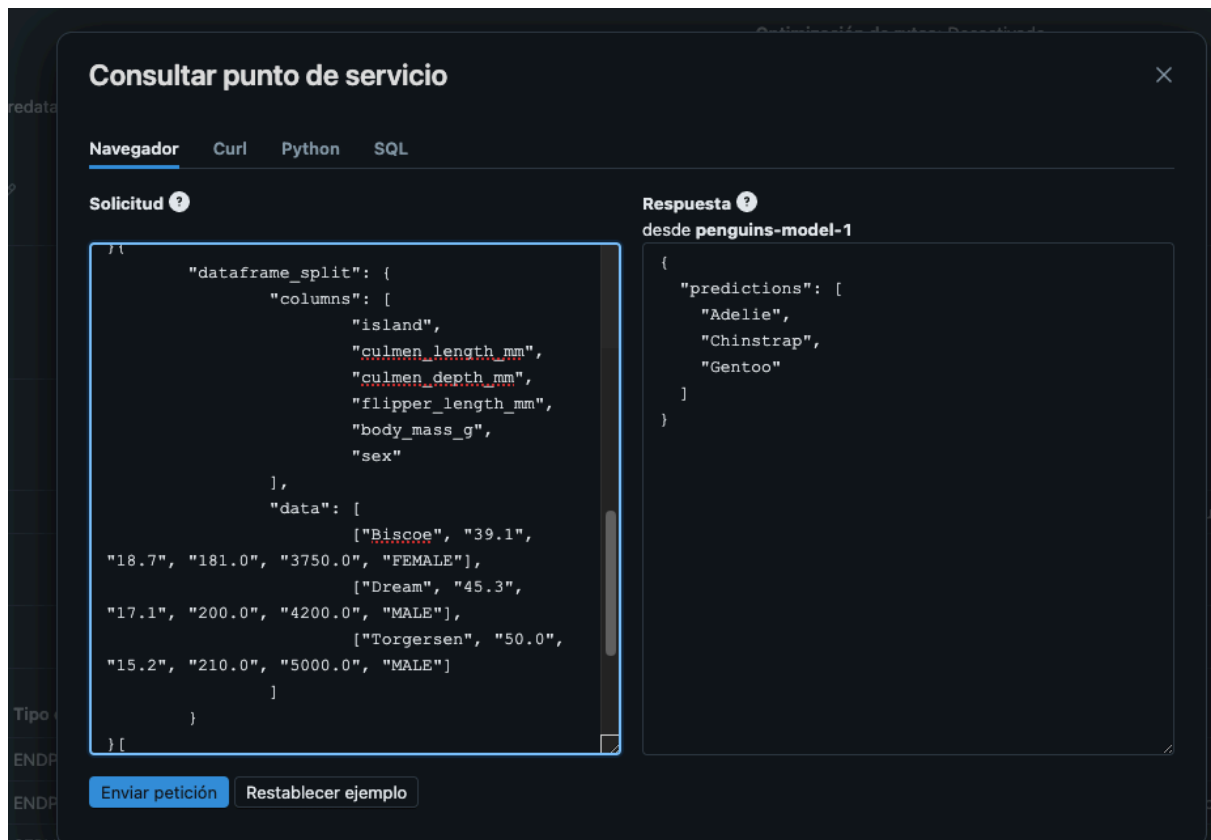
def score_model(dataset):
    url = 'https://adb-1942207080158662.2.azuredatabricks.net/serving-endpoints/penguin/invocations'
    headers = {'Authorization': f'Bearer {os.environ.get("DATABRICKS_TOKEN")}', 'Content-Type': 'application/json'}
    ds_dict = {'dataframe_split': dataset.to_dict(orient='split')} if isinstance(dataset, pd.DataFrame) else create_tf_serving_json(dataset)
    data_json = json.dumps(ds_dict, allow_nan=True)
    response = requests.request(method='POST', headers=headers, url=url, data=data_json)
    if response.status_code != 200:
        raise Exception(f'Request failed with status {response.status_code}, {response.text}')
    return response.json()
```

Aquí dejo una tabla comparando resultados de la práctica 4 y de las predicciones de este modelo.

Nº	Predicción práctica 4 (correcto)	Predicción con AutoML
1	Adelie	Gentoo
2	Gentoo	Gentoo
3	Chinstrap	Chinstrap
4	Chinstrap	Adelie
5	Gentoo	Gentoo

A pesar de que algunos resultados coinciden, parece que no ha conseguido acertar en todas las predicciones.

Cómo última prueba realizaré la misma petición que mi compañero Carlos Sánchez, de esta manera tengo la manera de comparar resultados para una misma petición.



Las predicciones sí coinciden con las de Carlos, esto me da a entender que el tiempo de entrenamiento no ha sido suficiente para obtener un modelo con un funcionamiento perfecto. Esa es justo la observación que se podría poner en práctica en caso de repetir este ejercicio para conseguir mejores resultados.