

Challenge AWS AI Practitioner Certified



AWS COMMUNITY
CARTAGENA

BIENVENIDOS



Semana 3: Desarrollando Aplicaciones de machine learning e inteligencia artificial



AWS COMMUNITY
CARTAGENA

Acerca de mí



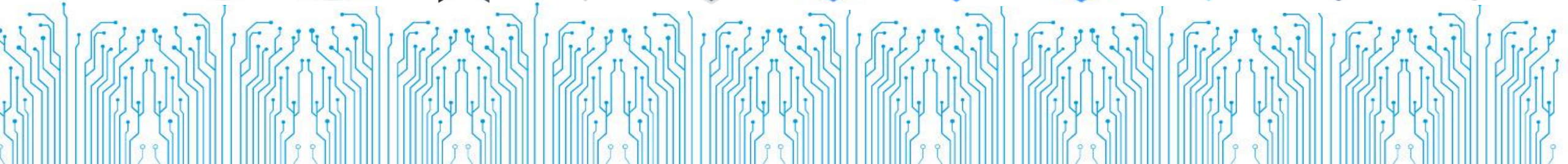
CloudOps Guild
Together, towards mastery in Cloud and DevOps



Esposo y padre de 2 hijos, Ángel y Sara
Ingeniero de sistemas, especialista en telecomunicaciones, MBA
Trabajo en el campo de la tecnología hace más de 20 años.

Áreas de especialización

- Arquitectura TI
- DevOps
- Site Reliability Engineering
- Arquitectura multi cloud (AWS, Azure, OCI)
- <https://www.linkedin.com/in/mario-rodrigo-serrano-pineda/>
- E-mail: marosepi2020@gmail.com
- <https://medium.com/@marioserranopineda>



AGENDA

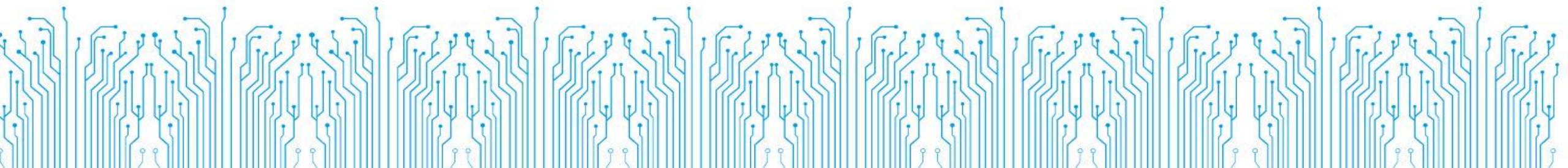


CloudOps Guild
Together, towards mastery in Cloud and DevOps

1. Desarrollando aplicaciones de machine learning
2. Desarrollando aplicaciones de inteligencia artificial generativa
3. Ingeniería de prompts



AWS COMMUNITY
CARTAGENA





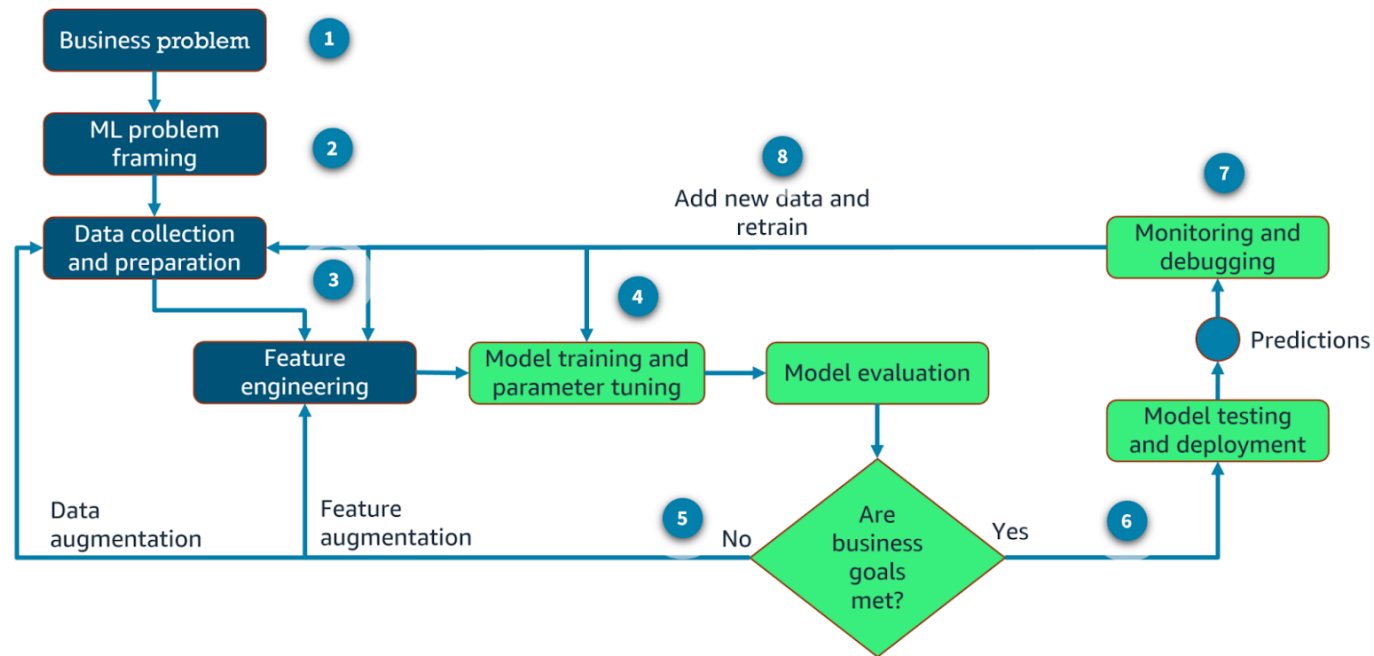
AWS COMMUNITY
CARTAGENA

1. Desarrollando aplicaciones de machine learning

Ciclo de vida del desarrollo de Machine Learning



CloudOps Guild
Together, towards mastery in Cloud and DevOps



- Business goal identification
- ML problem framing
- Data processing (data collection, data preprocessing, and feature engineering)
- Model development (training, tuning, and evaluation)
- Model retraining
- Model deployment (inference and prediction)
- Model monitoring

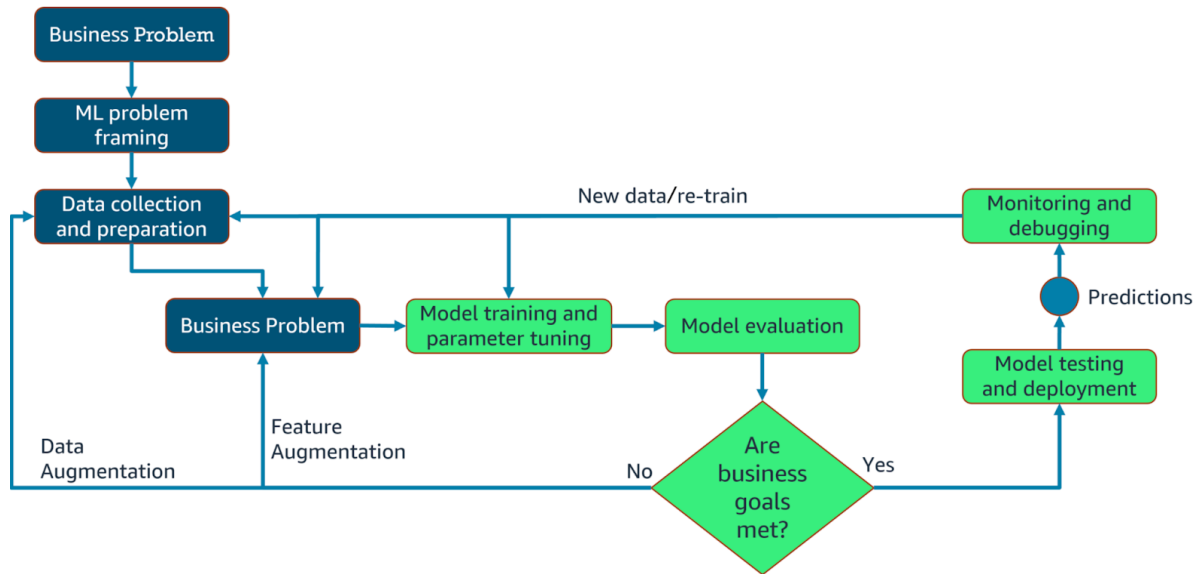
El ciclo de vida del aprendizaje automático (ML) se refiere al proceso completo de desarrollo, implementación y mantenimiento de modelos de aprendizaje automático.



AWS COMMUNITY
CARTAGENA



Business goal identification



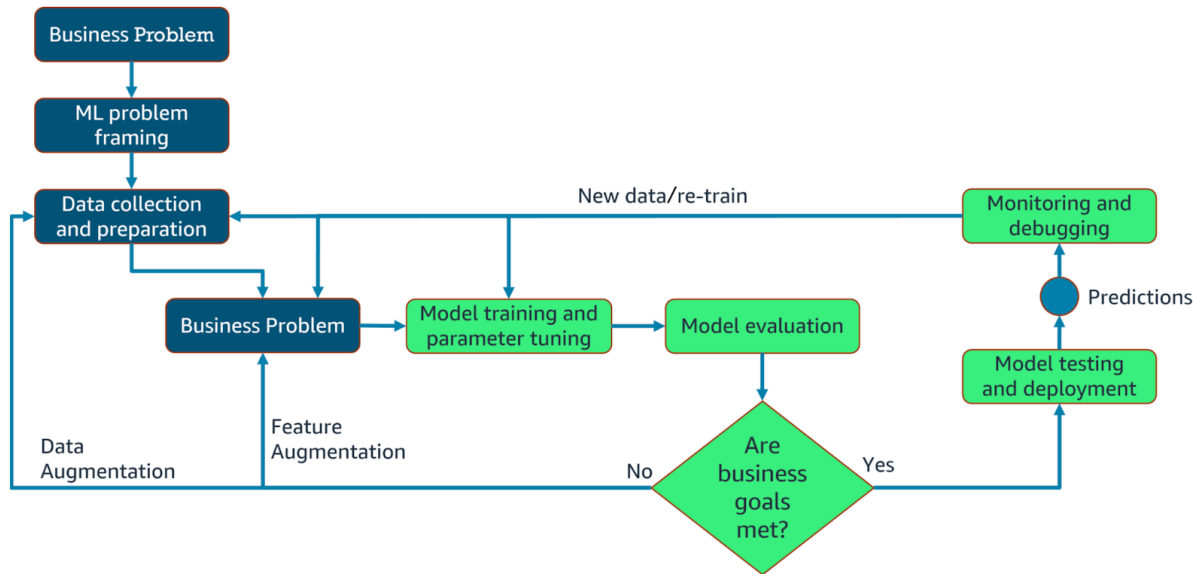
Definir los objetivos empresariales

El aprendizaje automático (ML) comienza con un objetivo empresarial. Los interesados en el negocio definen el valor, el presupuesto y los criterios de éxito. Definir los criterios de éxito o indicadores clave de rendimiento (KPIs) para la carga de trabajo de ML es fundamental.





ML problem framing



Formulación del problema de ML

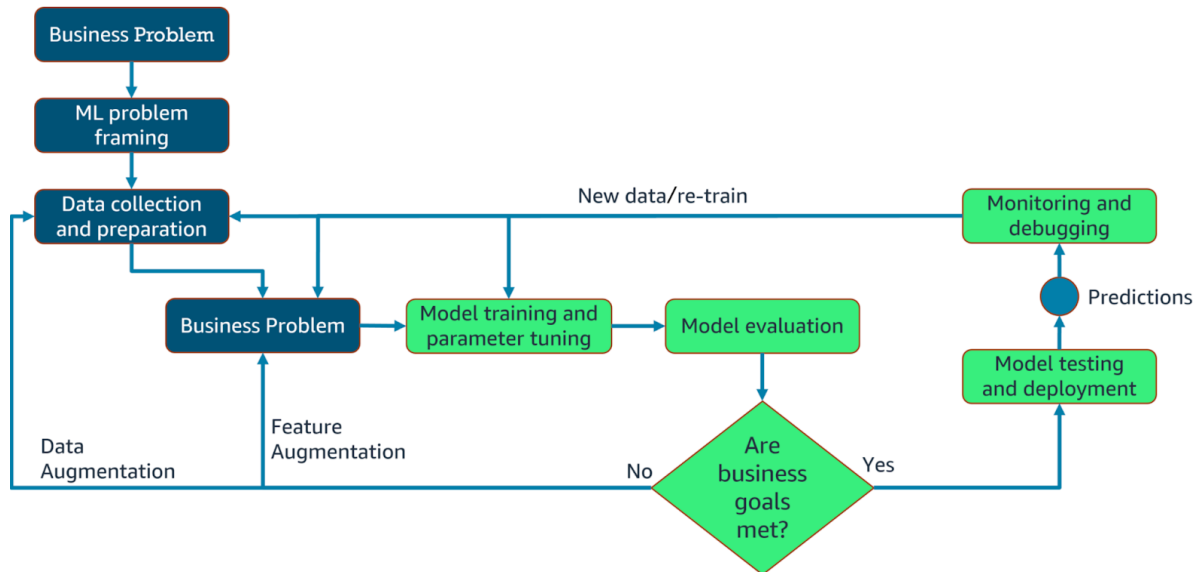
La formulación del problema implica articular el problema empresarial y convertirlo en un problema de aprendizaje automático.

Los científicos de datos, ingenieros de datos y arquitectos de ML colaboran con los expertos en la materia del negocio (SMEs) para determinar si el uso de ML es adecuado para resolver el problema empresarial. En esta fase, los equipos pueden trabajar en la fase de descubrimiento. Evaluarán si disponen de los datos, habilidades y recursos adecuados para desarrollar con éxito la solución empresarial.





Data processing (data collection, data preprocessing, and feature engineering)



Procesamiento de datos

Una vez formulado el problema, la siguiente fase es la preparación y preprocesamiento de los datos.

Para entrenar un modelo de ML preciso, los desarrolladores utilizan el procesamiento de datos para convertir la información en un formato utilizable.

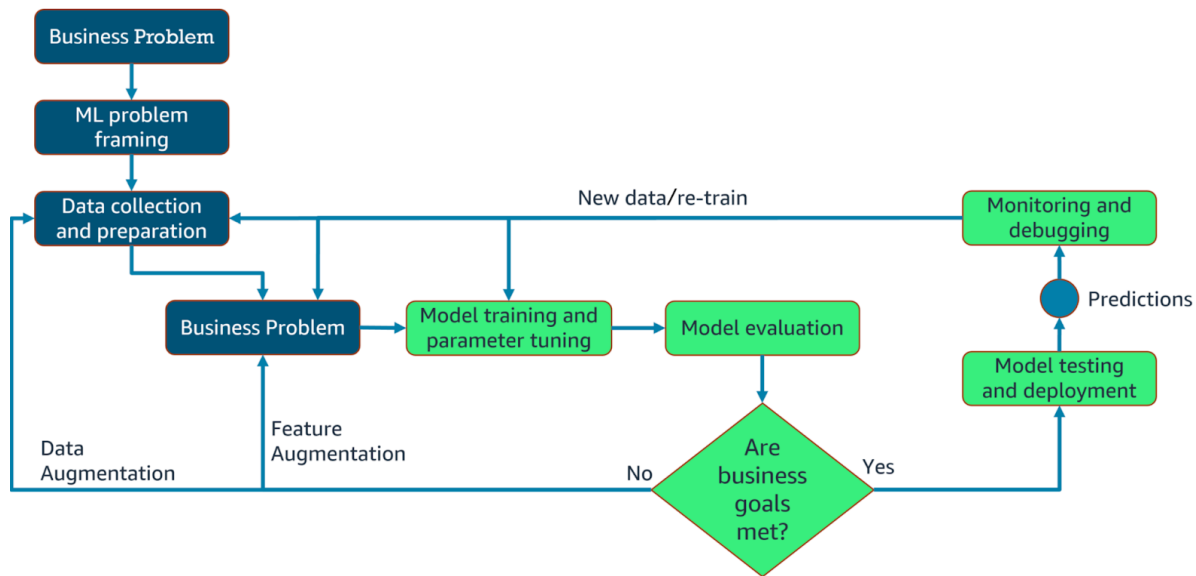
Los pasos del procesamiento de datos incluyen:

- **Recolección e integración de datos:** Garantiza que los datos en bruto estén disponibles en un lugar centralizado y accesible.
- **Preprocesamiento y visualización de datos:** Transforma los datos en bruto en un formato comprensible.
- **Ingeniería de características:** Consiste en crear, transformar, extraer y seleccionar variables a partir de los datos.





Model development (training, tuning, and evaluation)



Desarrollo del modelo

El desarrollo del modelo incluye el entrenamiento, ajuste y evaluación del modelo.

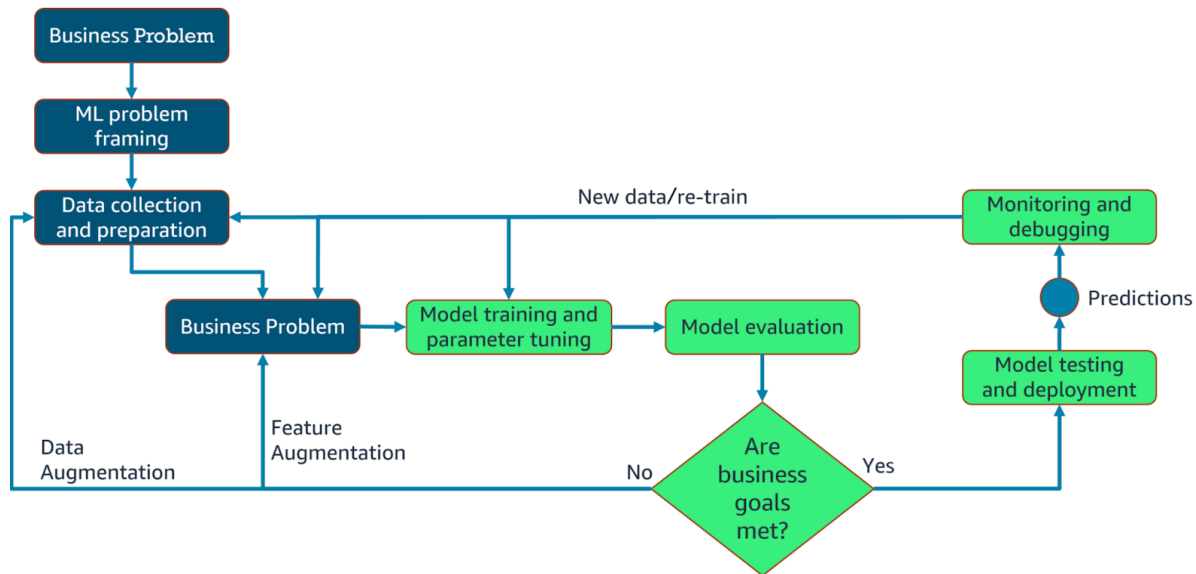
Es un proceso iterativo que se puede realizar varias veces a lo largo del flujo de trabajo.

Inicialmente, al entrenar el modelo, es probable que los resultados no sean los esperados. Por lo tanto, los desarrolladores pueden realizar ingeniería de características adicional y ajustar los hiperparámetros del modelo antes de volver a entrenarlo.





Model retraining



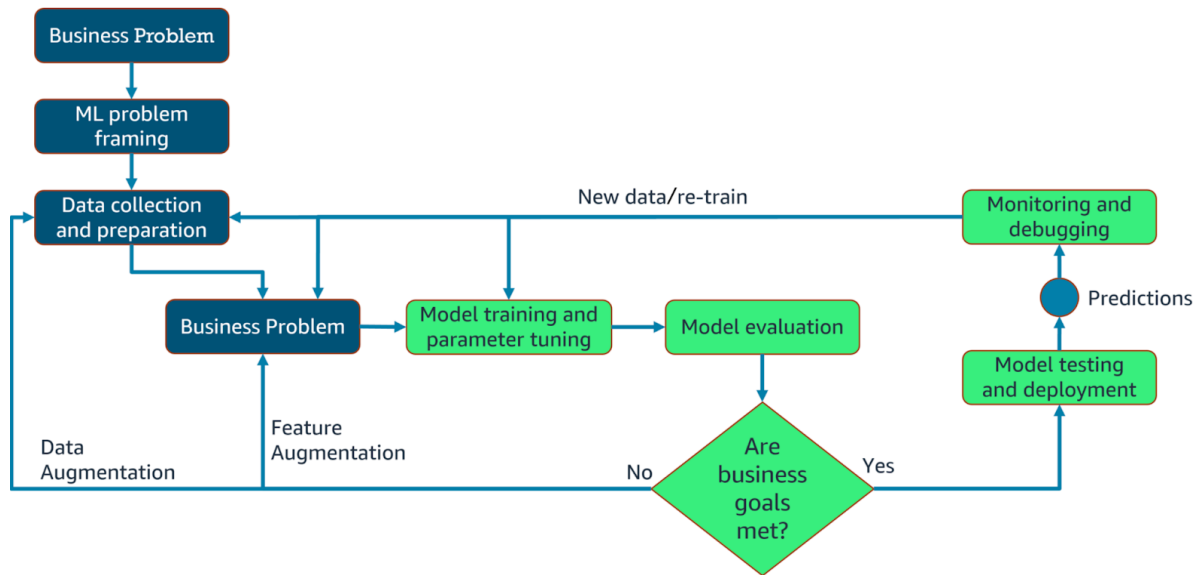
Reentrenamiento

Si el modelo no cumple con los objetivos empresariales, es necesario revisar los datos y las características para identificar formas de mejorarlo. La construcción de un modelo suele ser un proceso iterativo. También puede implicar ajustes en los hiperparámetros de entrenamiento.





Model deployment (inference and prediction)



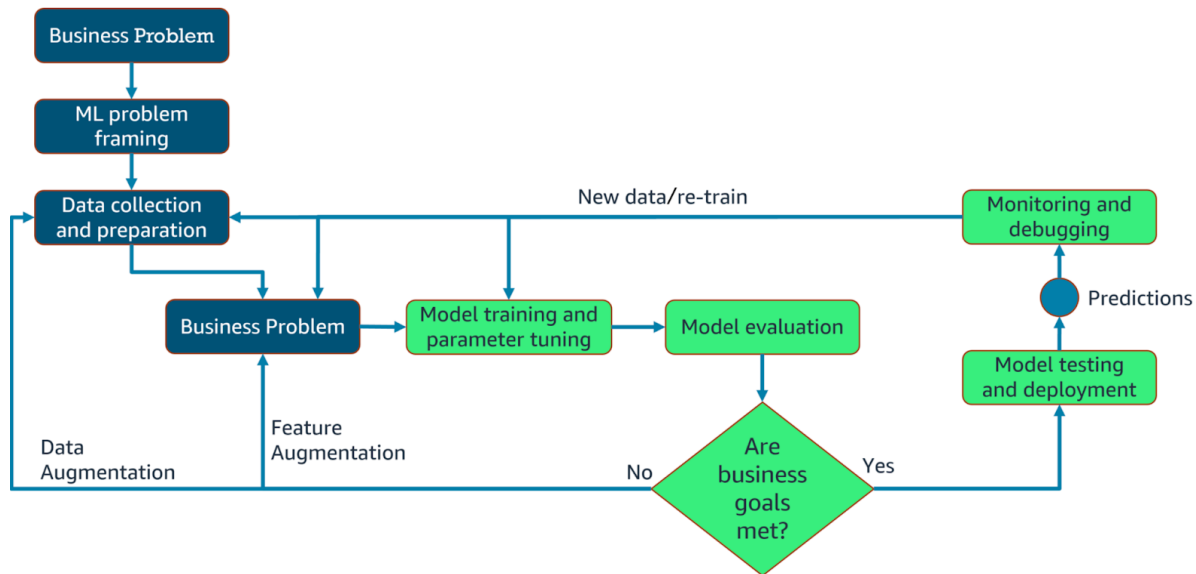
Despliegue

Si los resultados son satisfactorios, el modelo se implementa en producción. En este punto, el modelo está listo para realizar predicciones e inferencias.





Model monitoring



Monitoreo

El sistema de monitoreo del modelo garantiza que mantenga un nivel de rendimiento adecuado mediante la detección y mitigación temprana de problemas. También ayuda a depurar errores y a comprender el comportamiento del modelo.



Desarrollo de soluciones de ML con Amazon SageMaker



CloudOps Guild
Together, towards mastery in Cloud and DevOps



Amazon SageMaker es un servicio de aprendizaje automático (ML) totalmente gestionado. A través de una única interfaz visual unificada, puedes realizar las siguientes tareas:

- **Recolectar y preparar datos.**
- **Construir y entrenar modelos de aprendizaje automático.**
- **Desplegar los modelos y monitorear el rendimiento de sus predicciones.**



AWS COMMUNITY
CARTAGENA



Recolección, análisis y preparación de datos

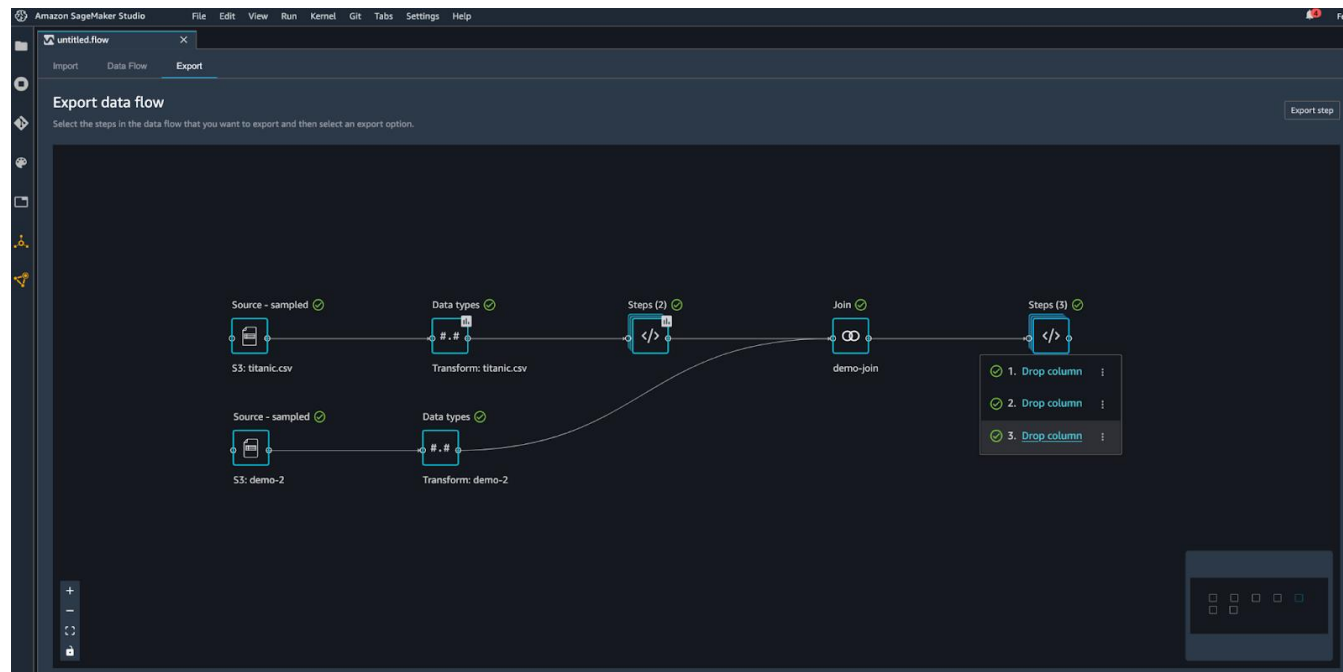


CloudOps Guild
Together, towards mastery in Cloud and DevOps

Amazon SageMaker Data Wrangler es una herramienta de **bajo código y sin código (LCNC)**. Proporciona una solución integral para importar, preparar, transformar, crear características y analizar datos mediante una interfaz web. Además, los usuarios pueden agregar sus propios scripts y transformaciones en **Python** para personalizar los flujos de trabajo.

Para usuarios más avanzados y necesidades de preparación de datos a gran escala, **Amazon SageMaker Studio Classic** cuenta con integración nativa con **Amazon EMR** y **AWS Glue interactive sessions**, lo que permite manejar la preparación de datos y los flujos de trabajo de ML dentro de un **notebook de SageMaker Studio Classic**

Por último, utilizando la **SageMaker Processing API**, los clientes pueden ejecutar scripts y notebooks para **procesar, transformar y analizar conjuntos de datos**. También pueden emplear diversos frameworks de ML como **scikit-learn**, **MXNet** o **PyTorch**, aprovechando entornos de aprendizaje automático completamente gestionados. Al final de esta fase, los clientes obtienen características definidas para el modelo y los datos preparados para su entrenamiento.



Amazon SageMaker Clarify ayuda a mejorar la calidad de los modelos proporcionando herramientas para:

- **Detección de sesgos** durante la preparación de datos y después del entrenamiento del modelo.
- **Explicabilidad del modelo**, generando informes que permiten a los interesados comprender **cómo y por qué** los modelos realizan determinadas predicciones.



AWS COMMUNITY
CARTAGENA

Managing features

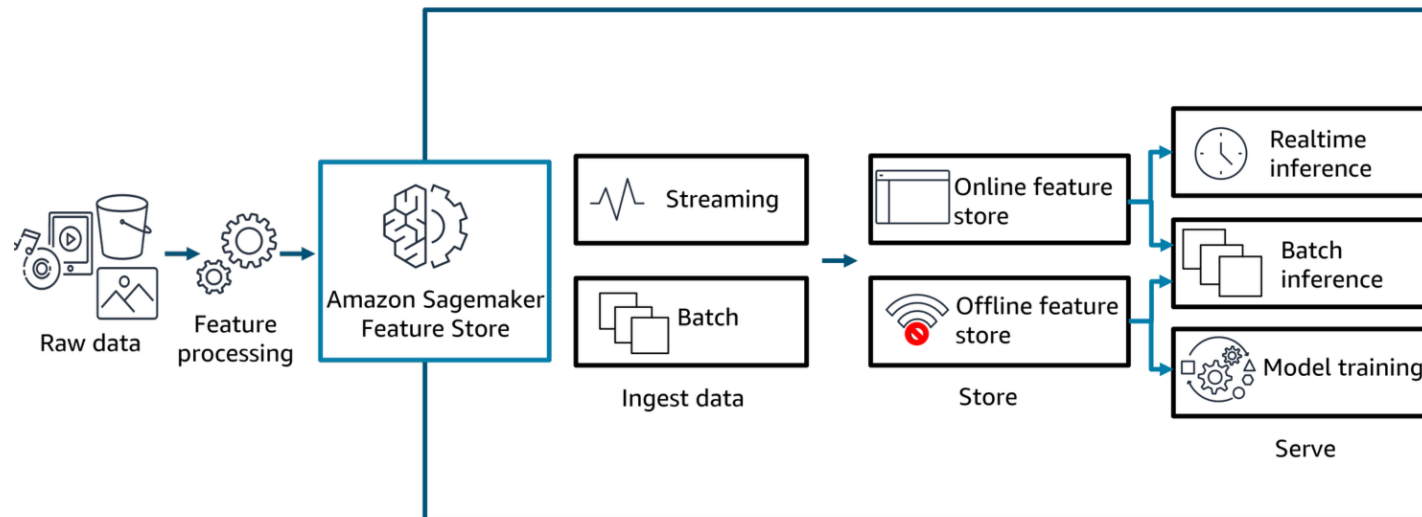


CloudOps Guild
Together, towards mastery in Cloud and DevOps

Amazon SageMaker Feature Store

Ayuda a científicos de datos, ingenieros de aprendizaje automático y profesionales en general a **crear, compartir y gestionar características** para el desarrollo de modelos de ML.

Las características almacenadas en **Feature Store** pueden ser recuperadas y enriquecidas antes de ser utilizadas en los modelos de ML para realizar **inferencias**.



AWS COMMUNITY
CARTAGENA

Model training and evaluation



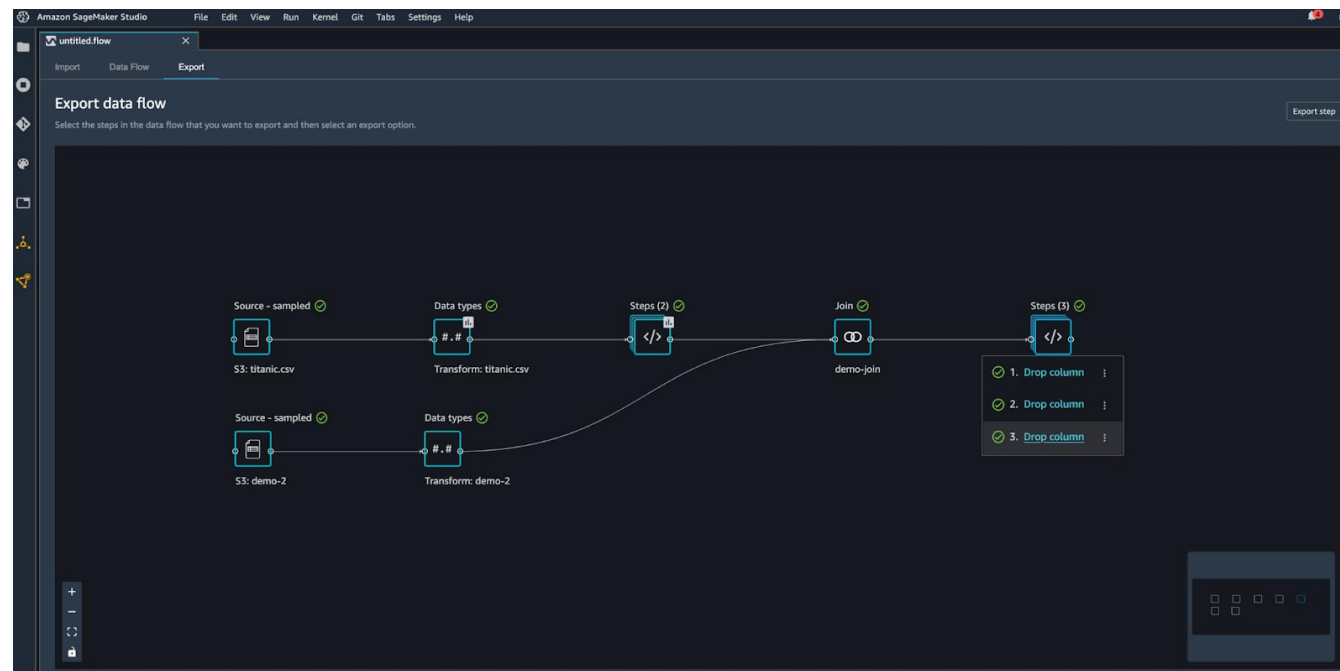
CloudOps Guild
Together, towards mastery in Cloud and DevOps

Amazon SageMaker Data Wrangler es una herramienta de **bajo código y sin código (LCNC)**. Proporciona una solución integral para importar, preparar, transformar, crear características y analizar datos mediante una interfaz web. Además, los usuarios pueden agregar sus propios scripts y transformaciones en **Python** para personalizar los flujos de trabajo.

Para usuarios más avanzados y necesidades de preparación de datos a gran escala, **Amazon SageMaker Studio Classic** cuenta con integración nativa con **Amazon EMR** y **AWS Glue interactive sessions**, lo que permite manejar la preparación de datos y los flujos de trabajo de ML dentro de un **notebook de SageMaker Studio Classic**

Por último, utilizando la **SageMaker Processing API**, los clientes pueden ejecutar scripts y notebooks para **procesar, transformar y analizar conjuntos de datos**. También pueden emplear diversos frameworks de ML como **scikit-learn, MXNet o PyTorch**, aprovechando entornos de aprendizaje automático completamente gestionados.

Al final de esta fase, los clientes obtienen características definidas para el modelo y los datos preparados para su entrenamiento.



AWS COMMUNITY
CARTAGENA

Recolección, análisis y preparación de datos



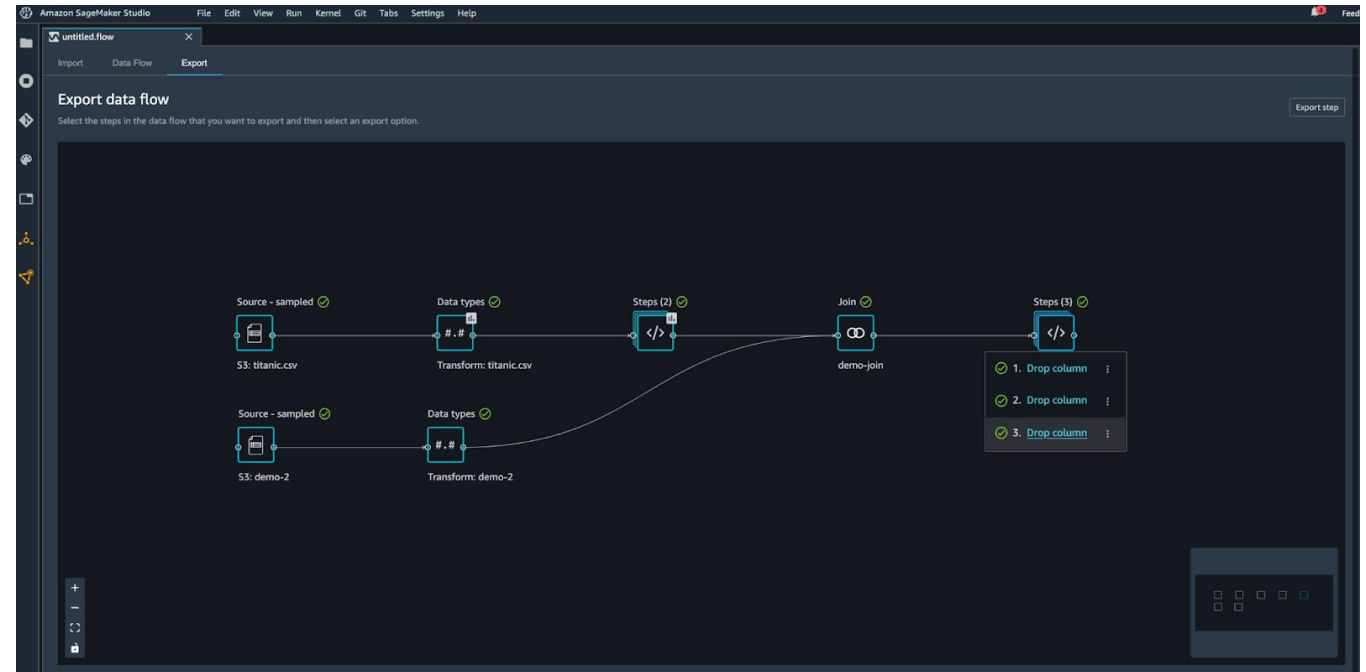
CloudOps Guild
Together, towards mastery in Cloud and DevOps

Amazon SageMaker Data Wrangler es una herramienta de **bajo código y sin código (LCNC)**. Proporciona una solución integral para importar, preparar, transformar, crear características y analizar datos mediante una interfaz web. Además, los usuarios pueden agregar sus propios scripts y transformaciones en **Python** para personalizar los flujos de trabajo.

Para usuarios más avanzados y necesidades de preparación de datos a gran escala, **Amazon SageMaker Studio Classic** cuenta con integración nativa con **Amazon EMR** y **AWS Glue interactive sessions**, lo que permite manejar la preparación de datos y los flujos de trabajo de ML dentro de un **notebook de SageMaker Studio Classic**

Por último, utilizando la **SageMaker Processing API**, los clientes pueden ejecutar scripts y notebooks para **procesar, transformar y analizar conjuntos de datos**. También pueden emplear diversos frameworks de ML como **scikit-learn**, **MXNet** o **PyTorch**, aprovechando entornos de aprendizaje automático completamente gestionados.

Al final de esta fase, los clientes obtienen características definidas para el modelo y los datos preparados para su entrenamiento.



AWS COMMUNITY
CARTAGENA

Entrenamiento y despliegue de modelos con Amazon SageMaker



CloudOps Guild
Together, towards mastery in Cloud and DevOps

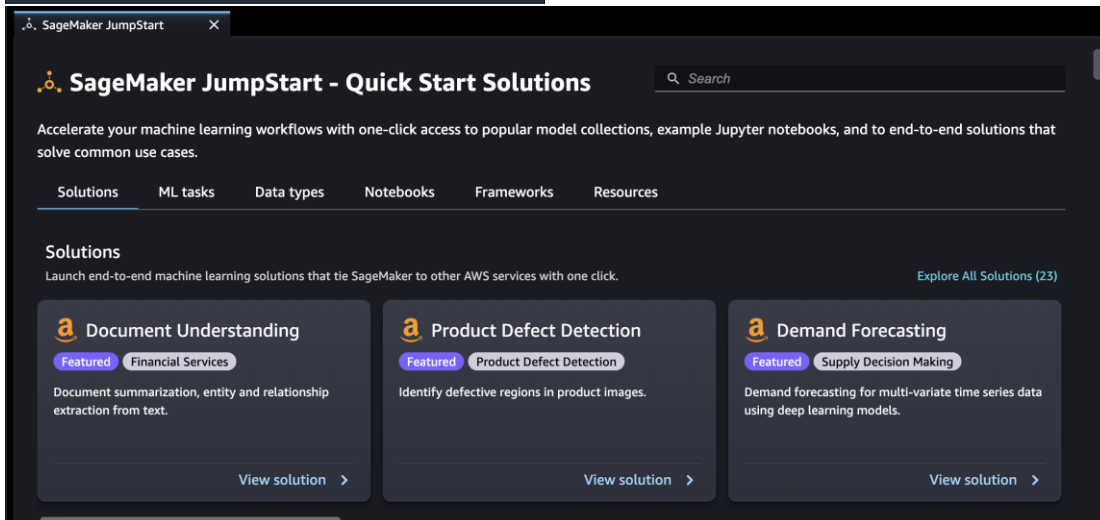


Amazon SageMaker Canvas

Amazon SageMaker ofrece la funcionalidad de **trabajos de entrenamiento** para entrenar y desplegar modelos utilizando **algoritmos integrados o personalizados**.

Opciones de entrenamiento sin código o con código mínimo (LCNC)

- **Amazon SageMaker Canvas** permite a los clientes utilizar machine learning para generar predicciones sin necesidad de escribir código.
- **Amazon SageMaker JumpStart** proporciona **modelos preentrenados y de código abierto** que los clientes pueden utilizar para resolver una amplia variedad de problemas.



AWS COMMUNITY
CARTAGENA

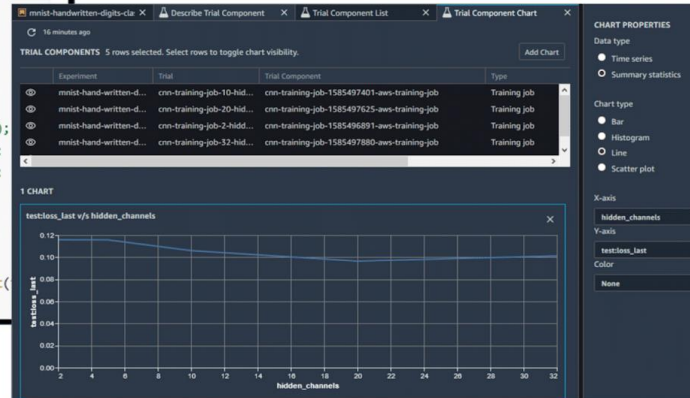
Model evaluation



CloudOps Guild
Together, towards mastery in Cloud and DevOps

```
estimator = PyTorch(
    py_version='py3',
    framework_version='1.1.0',
    ...,
    hyperparameters={
        'hidden_channels': num_hidden_channel,
        ...
    },
    metric_definitions=[
        {'Name': 'train:loss', 'Regex': 'Train Loss: (.*)'},
        {'Name': 'test:loss', 'Regex': 'Test Average loss: (.*)'},
        {'Name': 'test:accuracy', 'Regex': 'Test Accuracy: (.*)'}
    ],
    enable_sagemaker_metrics=True,
)

cnn_training_job_name = "cnn-training-job-{}".format(int(
```



Los clientes pueden utilizar **Amazon SageMaker Experiments** para probar múltiples combinaciones de **datos, algoritmos y parámetros**, mientras observan el impacto de los cambios incrementales en la precisión del modelo.

Ajuste de hiperparámetros

El ajuste de hiperparámetros es una técnica para encontrar la mejor versión de un modelo. **Amazon SageMaker Automatic Model Tuning** optimiza este proceso ejecutando múltiples trabajos con diferentes combinaciones de hiperparámetros y evaluando cada uno según una métrica seleccionada por el usuario.



AWS COMMUNITY
CARTAGENA

Model evaluation



CloudOps Guild
Together, towards mastery in Cloud and DevOps

Esta separación permite evaluar el modelo de manera efectiva y garantizar que pueda realizar predicciones precisas en datos no vistos previamente.



Use this data to
improve the model



Use this data to
evaluate the
predictive quality
of the trained
model



AWS COMMUNITY
CARTAGENA

Métricas de Evaluación de Modelos de Machine Learning



CloudOps Guild
Together, towards mastery in Cloud and DevOps

Métricas para Clasificación

Estas métricas se usan cuando el modelo predice categorías (por ejemplo, "spam" o "no spam").

1. Accuracy (Exactitud)

Mide qué tan a menudo el modelo hace predicciones correctas.
Se calcula como:

$$Accuracy = \frac{\text{Número de predicciones correctas}}{\text{Total de predicciones}}$$

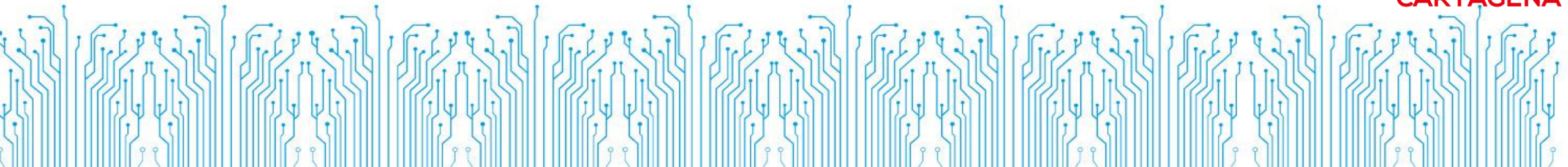
2. Precision (Precisión)

Indica qué porcentaje de las predicciones positivas realmente son positivas.

$$Precision = \frac{\text{Verdaderos Positivos (TP)}}{\text{Verdaderos Positivos (TP) + Falsos Positivos (FP)}}$$



AWS COMMUNITY
CARTAGENA



Métricas de Evaluación de Modelos de Machine Learning



CloudOps Guild
Together, towards mastery in Cloud and DevOps

3. Recall (Sensibilidad o Cobertura)

Mide qué porcentaje de los casos positivos fueron detectados correctamente.

$$Recall = \frac{\text{Verdaderos Positivos (TP)}}{\text{Verdaderos Positivos (TP)} + \text{Falsos Negativos (FN)}}$$

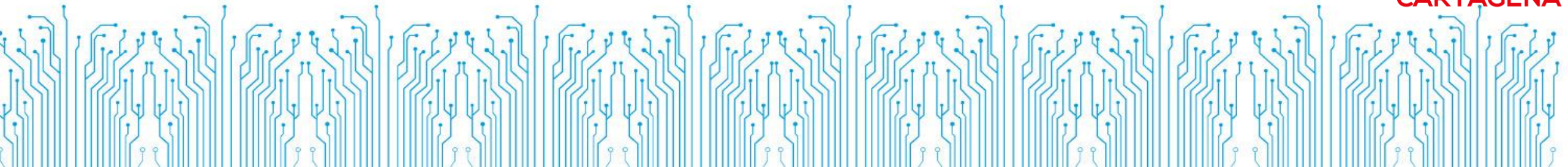
4. F1 Score

Es la media armónica entre **precisión** y **recall**, útil cuando hay un balance entre ambas métricas.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$



AWS COMMUNITY
CARTAGENA



Métricas de Evaluación de Modelos de Machine Learning



CloudOps Guild
Together, towards mastery in Cloud and DevOps

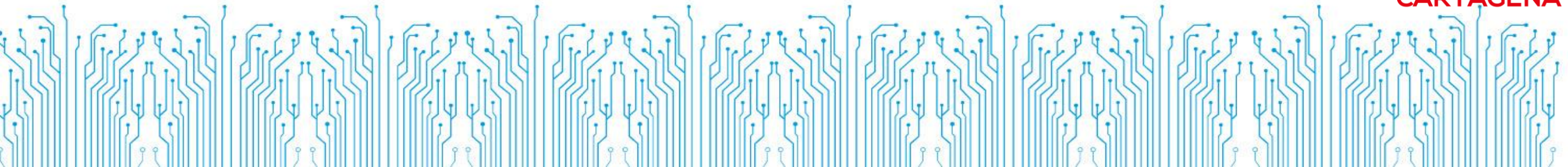
5. AUC-ROC (Área bajo la curva ROC)

Mide qué tan bien el modelo distingue entre clases positivas y negativas, considerando diferentes umbrales de decisión.

- **AUC (Área bajo la curva):** Valor entre **0 y 1**, donde **1** es un modelo perfecto y **0.5** es un modelo aleatorio.
- **ROC (Receiver Operating Characteristic Curve):** Gráfica que muestra la tasa de verdaderos positivos (**Recall**) contra la tasa de falsos positivos (**1 - Especificidad**).



AWS COMMUNITY
CARTAGENA



Métricas de Evaluación de Modelos de Machine Learning



CloudOps Guild
Together, towards mastery in Cloud and DevOps

Métricas para Regresión

Estas métricas evalúan modelos que predicen valores numéricos.

1. Mean Squared Error (MSE) - Error Cuadrático Medio

Mide el error promedio al cuadrado entre las predicciones y los valores reales.

$$MSE = \frac{1}{n} \sum (y_{\text{real}} - y_{\text{predicho}})^2$$

2. R Squared (R^2) - Coeficiente de Determinación

Mide qué porcentaje de la variabilidad de los datos es explicada por el modelo. Su valor varía entre **0 y 1**, donde **1** significa que el modelo explica **toda la variabilidad** de los datos.

$$R^2 = 1 - \frac{\sum (y_{\text{real}} - y_{\text{predicho}})^2}{\sum (y_{\text{real}} - \bar{y})^2}$$



AWS COMMUNITY
CARTAGENA

Métricas de Evaluación de Modelos de Machine Learning



Métrica	Explicación
Accuracy	% de predicciones correctas
Precision	% de positivos predichos correctamente
Recall	% de positivos detectados correctamente
F1 Score	Equilibrio entre precisión y recall
AUC-ROC	Capacidad de distinguir entre clases
MSE	Error promedio al cuadrado
R ²	% de la variabilidad explicada



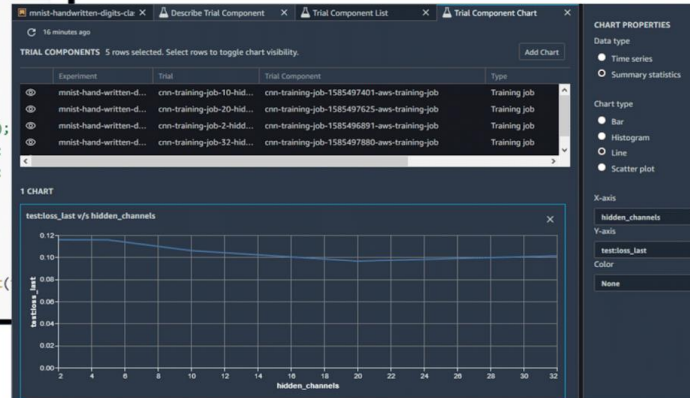
Deployment



CloudOps Guild
Together, towards mastery in Cloud and DevOps

```
estimator = PyTorch(
    py_version='py3',
    framework_version='1.1.0',
    ...,
    hyperparameters={
        'hidden_channels': num_hidden_channel,
        ...
    },
    metric_definitions=[
        {'Name': 'train:loss', 'Regex': 'Train Loss: (.*)'},
        {'Name': 'test:loss', 'Regex': 'Test Average loss: (.*)'},
        {'Name': 'test:accuracy', 'Regex': 'Test Accuracy: (.*)'}
    ],
    enable_sagemaker_metrics=True,
)

cnn_training_job_name = "cnn-training-job-{}".format(int(
```



Con **Amazon SageMaker**, los clientes pueden desplegar sus modelos de aprendizaje automático (ML) para realizar predicciones, también conocidas como **inferencias**. SageMaker ofrece una amplia variedad de opciones de **infraestructura de ML y despliegue de modelos** para satisfacer todas las necesidades de inferencia.



AWS COMMUNITY
CARTAGENA

Monitoring

Amazon SageMaker Model Monitor permite a los clientes **supervisar la calidad de los modelos** en producción.

Los usuarios pueden configurar:

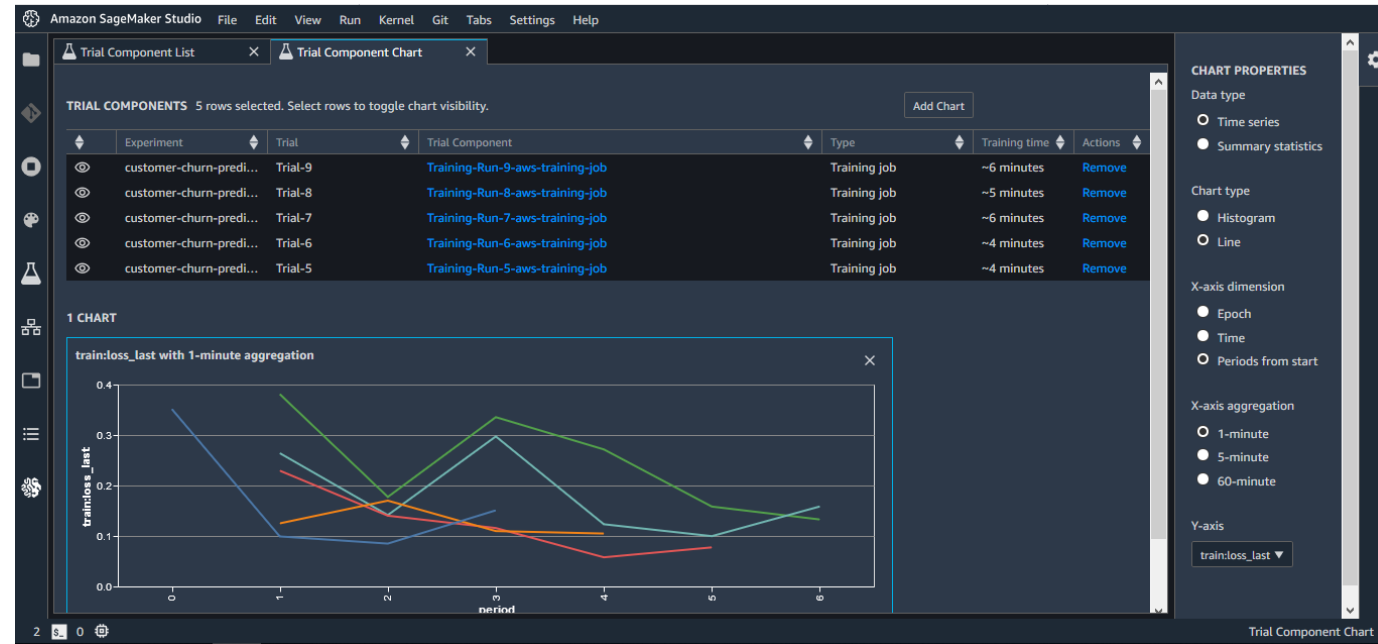
- **Monitoreo continuo**
- **Monitoreo programado**

SageMaker Model Monitor ayuda a mantener la calidad del modelo al detectar violaciones en los umbrales definidos por el usuario para:

- **Calidad de los datos**
- **Calidad del modelo**
- **Deriva de sesgo (bias drift)**
- **Deriva en la atribución de características (feature attribution drift)**



CloudOps Guild
Together, towards mastery in Cloud and DevOps



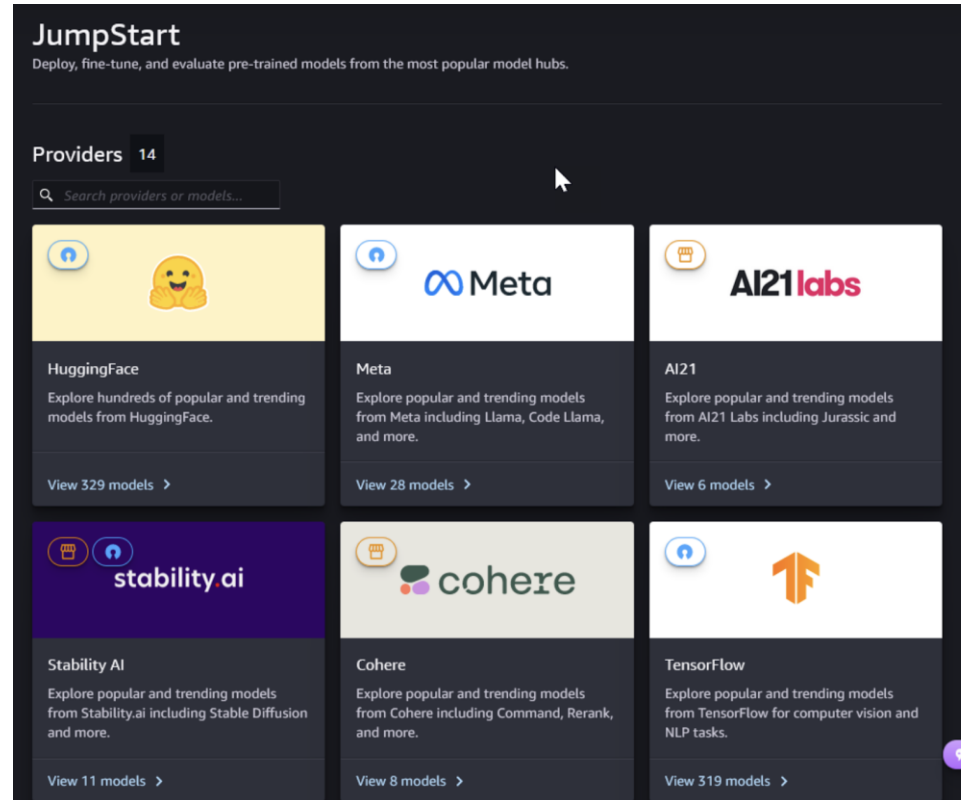
AWS COMMUNITY
CARTAGENA

SageMaker Jumpstart

Con **SageMaker JumpStart**, puedes **desplegar, ajustar y evaluar modelos preentrenados** de los principales repositorios de modelos.

Características principales

- Proporciona **modelos preentrenados de código abierto** de proveedores líderes para diversos tipos de problemas, facilitando el inicio en machine learning.
- Permite **entrenar y ajustar** estos modelos de forma incremental antes de su despliegue.
- Incluye **plantillas de solución** que configuran la infraestructura necesaria para casos de uso comunes.
- Ofrece **notebooks de ejemplo listos para ejecutar**, que sirven como guía para el uso de SageMaker en proyectos de aprendizaje automático.



CloudOps Guild
Together, towards mastery in Cloud and DevOps



AWS COMMUNITY
CARTAGENA



AWS COMMUNITY
CARTAGENA

2. Desarrollando aplicaciones de inteligencia artificial

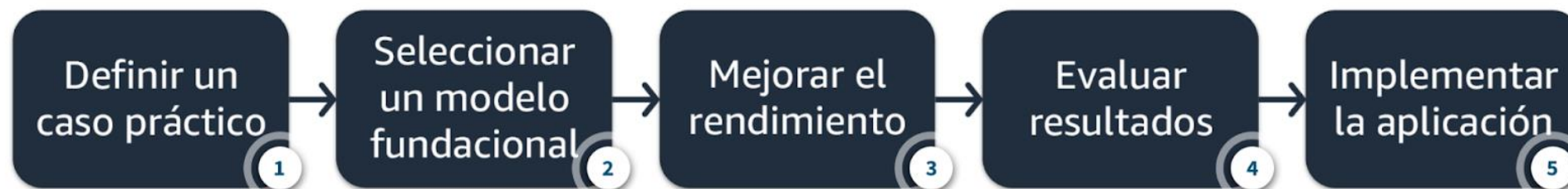


Ciclo de vida de las aplicaciones de IA generativa





Ciclo de vida de las aplicaciones de IA generativa



Definir un caso práctico

En la primera etapa, se identifican los requisitos para incorporar capacidades de IA generativa en una aplicación. Esto podría implicar el análisis de las funcionalidades de la aplicación, las necesidades de los usuarios y los objetivos empresariales para determinar dónde puede agregar valor la IA generativa.

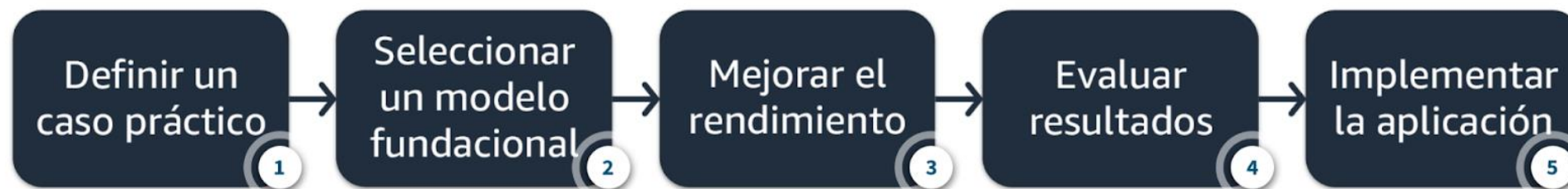
Seleccionar un modelo fundacional

En función de los requisitos identificados, se selecciona un modelo de IA generativa adecuado a partir de los modelos existentes previamente entrenados o se desarrolla desde cero. Esta decisión depende de factores como la disponibilidad de modelos previamente entrenados adecuados, la complejidad del caso práctico y la disponibilidad de datos específicos del dominio para el entrenamiento.





Ciclo de vida de las aplicaciones de IA generativa



Mejorar el rendimiento

El modelo de IA generativa seleccionado o desarrollado se integra en la infraestructura o base de código de la aplicación. Esto puede implicar la adaptación de los formatos de entrada y salida del modelo, el ajuste del modelo con datos específicos de la aplicación y la implementación de las personalizaciones u optimizaciones necesarias.

Evaluar resultados

Se realizan pruebas y evaluaciones exhaustivas de las capacidades de IA generativa integradas para garantizar que cumplen con los requisitos especificados y funcionan según lo esperado. Esto puede implicar la ejecución de pruebas con diversas entradas, casos extremos y situaciones del mundo real, así como la evaluación de la calidad, la coherencia y la relevancia del contenido generado.





Ciclo de vida de las aplicaciones de IA generativa



Implementación

Tras un resultado satisfactorio de las pruebas, la aplicación con capacidades de IA generativa integradas se implementa en el entorno de producción. Los mecanismos de supervisión se establecen para rastrear el rendimiento, el uso y los posibles problemas o sesgos asociados con los resultados del modelo de IA generativa.





AWS COMMUNITY
CARTAGENA

3. Ingeniería de Prompts



Ingeniería de peticiones

La ingeniería de peticiones se refiere al proceso de elaborar cuidadosamente las peticiones de entrada o las instrucciones que se dan al modelo para generar los resultados o comportamientos deseados. La redacción, la estructura y el contenido de la petición pueden influir significativamente en la calidad, la relevancia y las características de los resultados generados. La ingeniería de peticiones tiene como objetivo optimizar las peticiones para dirigir la generación del modelo en la dirección deseada, utilizando las capacidades del modelo y, al mismo tiempo, mitigando los posibles sesgos o los resultados no deseados.

¿Qué es prompt engineering?





Elementos de un prompt

La estructura de un **prompt** depende de la tarea que se le asigna a un modelo.

Instrucciones: Definen la tarea que debe realizar el modelo de lenguaje. Incluyen una descripción de la tarea o instrucciones sobre cómo debe ejecutarla.

•**Contexto:** Información externa que ayuda a guiar el modelo en su respuesta.

•**Datos de entrada:** Información proporcionada al modelo para obtener una respuesta.

•**Indicador de salida:** Define el tipo o formato de la respuesta esperada del modelo.

Escenario: Queremos que un modelo de lenguaje genere una descripción detallada de un producto para una tienda en línea.





Elementos de un prompt

Instrucciones:

Genera una descripción persuasiva y detallada para un producto en una tienda en línea. Asegúrate de incluir sus características principales y beneficios para los clientes.

Contexto:

Este producto es una **mochila impermeable para laptop de 15.6 pulgadas**, ideal para profesionales y estudiantes. La tienda en línea se especializa en productos tecnológicos y accesorios para el día a día.

Datos de entrada:

- **Producto:** Mochila impermeable para laptop
- **Características:** Material resistente al agua, múltiples compartimentos, puerto de carga USB, diseño ergonómico
- **Público objetivo:** Profesionales y estudiantes
- **Uso:** Trabajo, universidad, viajes

Indicador de salida:

El resultado debe ser un texto descriptivo atractivo, con un tono profesional y persuasivo, listo para usarse en la tienda en línea.









Elementos de un prompt

Salida esperada del modelo:

Mochila Impermeable para Laptop de 15.6" – Comodidad y Protección en Todo Momento

Descubre la mochila ideal para profesionales y estudiantes que buscan **comodidad, seguridad y estilo** en su día a día. Diseñada con **material impermeable de alta calidad**, protege tu laptop y documentos de la lluvia o derrames accidentales.

Características destacadas:

-  **Espaciosa y organizada:** Múltiples compartimentos para laptop, tablet, documentos y accesorios.
-  **Puerto USB integrado:** Carga tus dispositivos fácilmente mientras te desplazas.
-  **Diseño ergonómico:** Correas acolchadas y transpirables para mayor confort.
-  **Estilo moderno y profesional:** Perfecta para oficina, universidad o viajes.

¡Lleva tu productividad al siguiente nivel con esta mochila versátil y funcional!  





Modificación de Prompts

Parámetros de inferencia

Al interactuar con **modelos fundamentales (FMs)**, a menudo puedes **configurar parámetros de inferencia** para limitar o influenciar la respuesta del modelo.

Los parámetros disponibles dependen del modelo utilizado y suelen clasificarse en varias categorías. Las más comunes son:

1. **Aleatoriedad y diversidad**
2. **Longitud de la respuesta**

Aleatoriedad y diversidad

Esta es la categoría más común de parámetros de inferencia.

Los parámetros de **aleatoriedad y diversidad** afectan la **variación en las respuestas generadas**, limitando las salidas a las más probables o modificando la **distribución de probabilidad de las respuestas**.

Entre los parámetros más utilizados se encuentran:

- **Temperature**
- **Top-K**
- **Top-P**





Modificación de Prompts

Temperature (Temperatura)

Este parámetro **controla la aleatoriedad o creatividad** de la respuesta del modelo.

- **Mayor temperatura (ejemplo: 1.0):** Salidas más diversas, creativas e impredecibles, aunque pueden ser menos coherentes o relevantes.

- **Menor temperatura (ejemplo: 0.2):** Respuestas más conservadoras, repetitivas y enfocadas en las opciones más probables.

 **Rango:** Se ajusta entre **0 y 1**.

Top-P

Top-P (también llamado **nucleus sampling**) **controla la diversidad del texto** limitando el número de palabras elegibles según su probabilidad.

Se ajusta en una escala de **0 a 1**, donde:

- **Valores bajos (ejemplo: 0.2):** El modelo selecciona solo las palabras con mayor probabilidad, resultando en respuestas más predecibles y precisas.

- **Valores altos (ejemplo: 0.9):** Se amplía la variedad de palabras seleccionadas, permitiendo respuestas más creativas y variadas, pero potencialmente menos precisas.





Modificación de Prompts

Top-K

Top-K es un parámetro que **limita la cantidad de palabras** entre las cuales el modelo puede elegir la siguiente palabra en una secuencia, basándose únicamente en las **K palabras más probables**.

Por ejemplo:

- Si **Top-K = 50**, el modelo solo considerará **las 50 palabras más probables** para generar la siguiente palabra en la respuesta, **sin importar** qué porcentaje de probabilidad representan en la distribución total.

Esto permite controlar la **diversidad y coherencia** en las respuestas:

- **Valores bajos (ejemplo: 10):** Las respuestas serán más predecibles y estructuradas.
- **Valores altos (ejemplo: 100):** Se incluirán palabras menos frecuentes, lo que genera más diversidad, pero con el riesgo de obtener respuestas menos precisas.





Técnicas de Ingeniería de Prompts

Zero-shot prompting (Prompting sin ejemplos previos)

Es una técnica en la que el usuario presenta una tarea a un **modelo generativo sin proporcionar ejemplos previos ni entrenamiento explícito** para esa tarea en particular.

Few-shot prompting (Prompting con pocos ejemplos)

Consiste en proporcionar al modelo ejemplos de contexto que lo guíen en la **comprensión de la tarea y en la generación de la respuesta esperada**.

- **One-shot prompting:** Se proporciona **un solo ejemplo** para ayudar al modelo a entender la tarea.

- **Few-shot prompting:** Se proporcionan **varios ejemplos** para mejorar el desempeño del modelo.

Este enfoque ayuda a que el modelo genere respuestas más precisas al **condicionarlo con ejemplos previos**.

Chain-of-thought prompting (Prompting con razonamiento en cadena)

Es una técnica que **divide tareas de razonamiento complejas en pasos intermedios más pequeños**.

Para activar el **razonamiento en cadena**, se puede utilizar la frase:

→ **"Piensa paso a paso."**

Se recomienda **usar CoT prompting** cuando la tarea **requiere múltiples pasos de procesamiento lógico** o una serie de inferencias antes de llegar a una respuesta final.





CloudOps Guild
Together, towards mastery in Cloud and DevOps

Recursos adicionales

[prompt-engineering](#)

[prompt-engineering-guidelines](#)



AWS COMMUNITY
CARTAGENA

Agradecimientos



CloudOps Guild
Together, towards mastery in Cloud and DevOps



Próxima sesión:

- Optimización de modelos fundacionales
- Seguridad, cumplimiento y gobierno de soluciones IA



**AWS COMMUNITY
CARTAGENA**