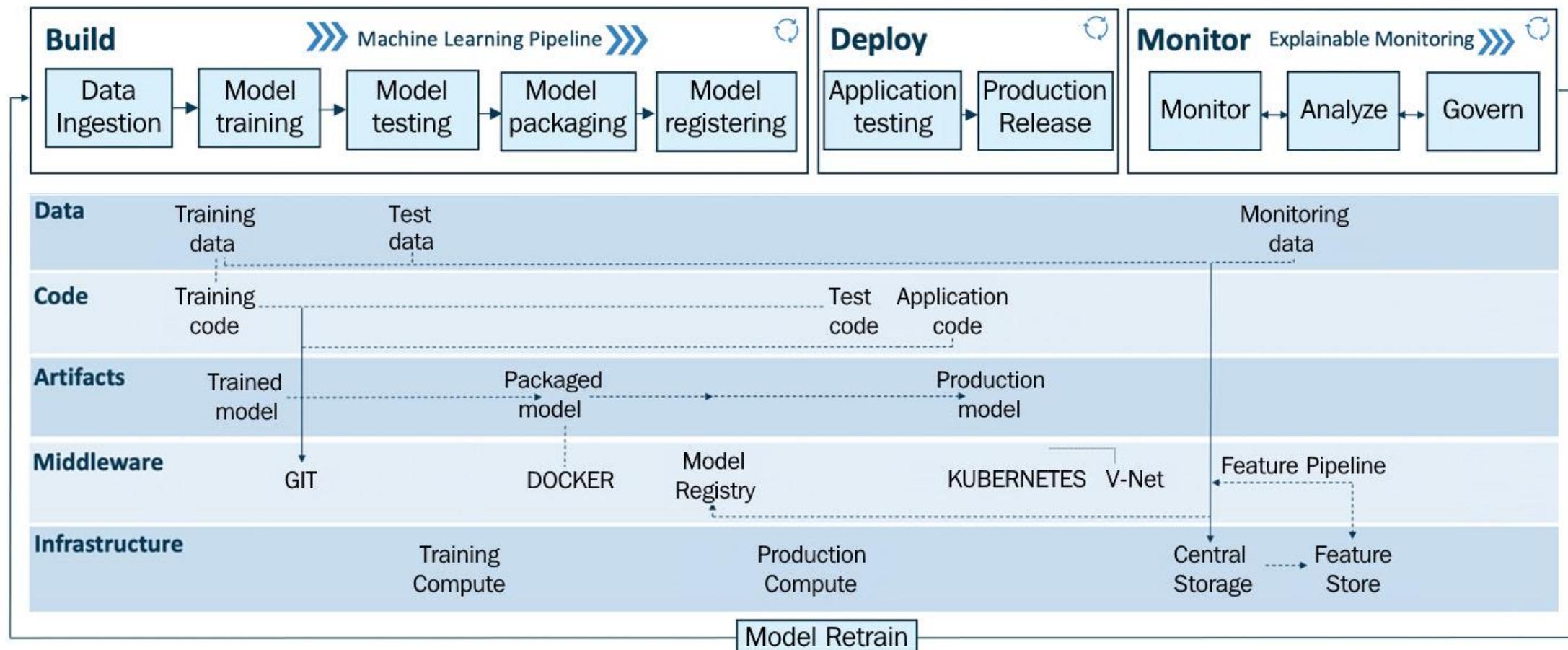


Conceptos y Flujo de trabajo de MLOps

1ra Parte

Flujo de Trabajo MLOps

MLOps Workflow

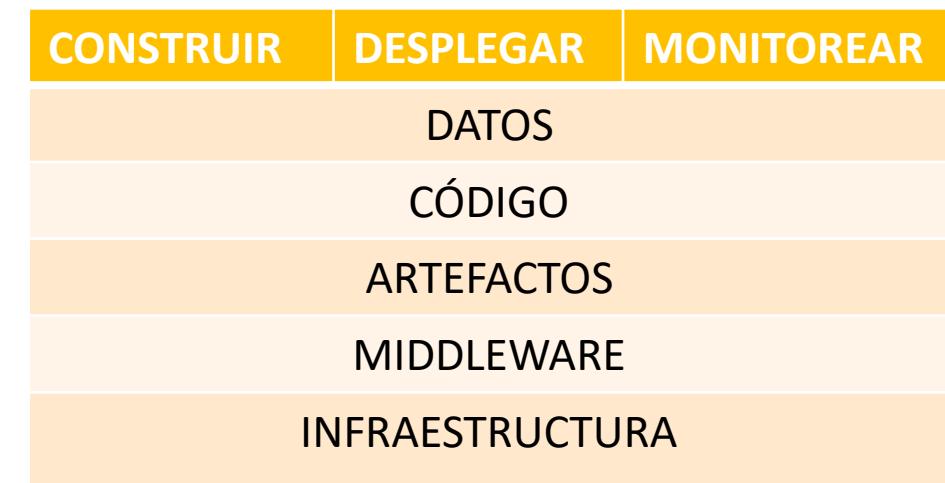


Flujo de Trabajo MLOps

- Este flujo de trabajo MLOps es genérico; es el resultado de muchas iteraciones del ciclo de diseño. Reúne ingeniería de datos, ML y DevOps de manera optimizada.
- Este flujo de trabajo genérico de MLOps; es modular y flexible y se puede utilizar para crear pruebas de concepto o para hacer operativas las soluciones de ML en cualquier negocio o industria.

Flujo de Trabajo de MLOps

- Este flujo de trabajo está segmentado en dos módulos:
 - Canalización de MLOps (construir, desplegar y monitorear): la capa superior
 - Controladores: datos, código, artefactos, middleware e infraestructura: capas medias e inferiores



Flujo de Trabajo MLOps

- La capa superior es la canalización de MLOps (construir, desplegar y monitorear), que está habilitada por controladores como datos, código, artefactos, middleware e infraestructura.
- La canalización de MLOps está impulsada por una variedad de servicios, controladores, middleware e infraestructura, y crea soluciones basadas en ML.
- Mediante el uso de esta canalización, una empresa o una persona puede crear prototipos, probar y validar rápidamente e implementar los modelos en producción a escala de manera frugal y eficiente.



CONSTRUIR

DESPLEGAR

MONITOREAR

Caso de Uso

- En este caso de uso, vamos a operacionalizar (prototipado y despliegue para la producción) un servicio de clasificación de imágenes de perros y gatos en un parque de mascotas. El servicio identificará perros y gatos en tiempo real a partir de los datos de inferencia provenientes de una cámara CCTV instalada en el parque de mascotas.



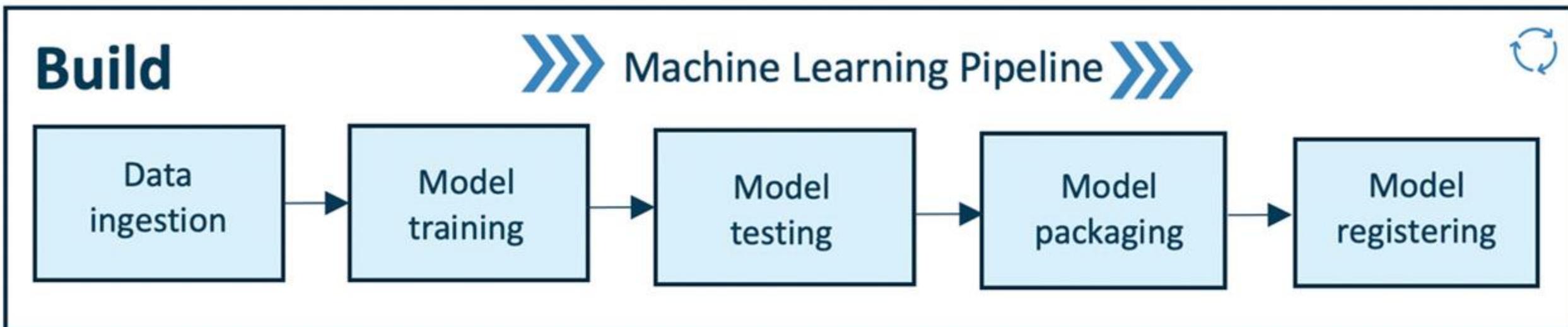
Caso de Uso

- El parque de mascotas le brinda acceso a los datos y la infraestructura necesarios para operar el servicio:
 - Datos: El parque de mascotas le ha dado acceso a su lago de datos que contiene 100 000 imágenes etiquetadas de perros y gatos, que se usarán para entrenar al modelo.
 - Infraestructura: Nube pública (IaaS).

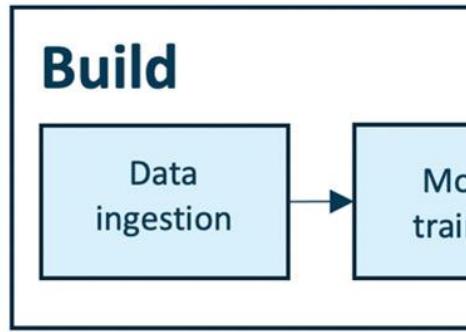


Construir

- El módulo de Construcción tiene la canalización principal de ML, y es exclusivamente para entrenar, empaquetar y crear versiones de los modelos de ML. Está soportado por los recursos de computación necesarios (por ejemplo, la CPU o la GPU en la nube o la computación distribuida) para ejecutar el entrenamiento y la canalización de ML:

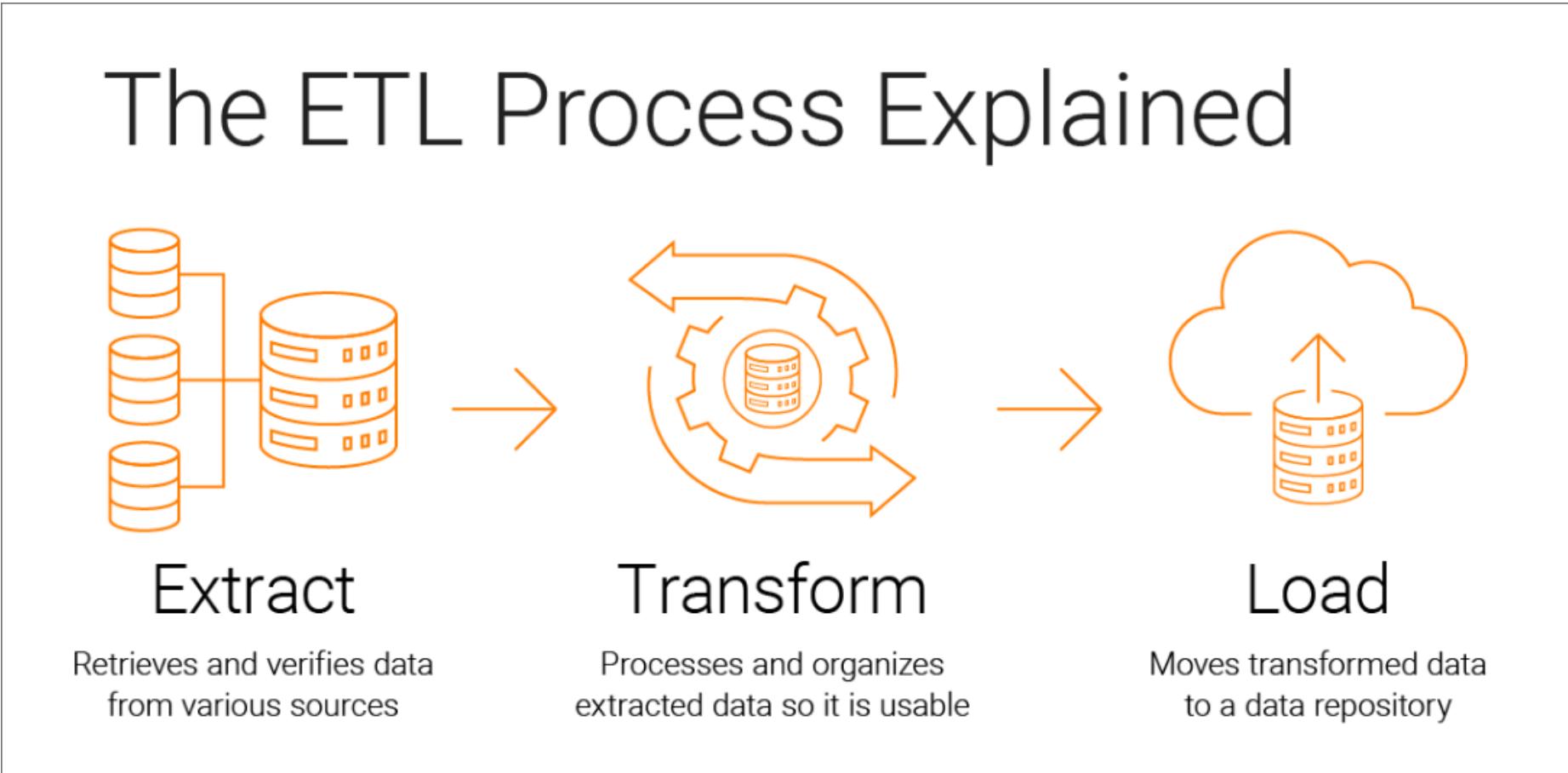


Ingestión de Datos



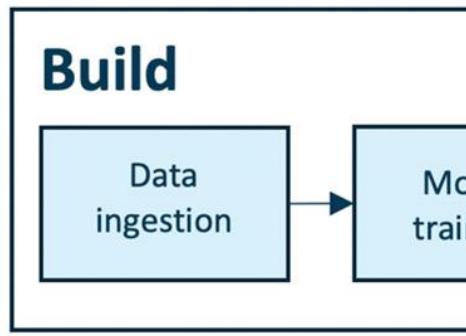
- Ingestión de datos: este paso desencadena la canalización de ML. Se ocupa del volumen, la velocidad, la veracidad y la variedad de datos mediante la **extracción de datos de varias fuentes de datos** (por ejemplo, bases de datos, data warehouses o lagos de datos) y la **ingesta** de los datos necesarios para el paso de **entrenamiento del modelo**.
- Las **canalizaciones de datos robustas** conectadas a múltiples fuentes de datos le permiten realizar operaciones de extracción, transformación y carga (**ETL**) para proporcionar los datos necesarios para fines de entrenamiento en ML.
- En este paso, podemos **dividir y versionar los datos** para el entrenamiento del modelo en el formato requerido (por ejemplo, el conjunto de entrenamiento o prueba).
- Como resultado de este paso, **cualquier experimento** (es decir, entrenamiento del modelo) **puede ser auditado y ser rastreable**.

Ingestión de Datos



Fuente: <https://www.informatica.com/resources/articles/what-is-etl.html>

Ingestión de Datos



Implementación del caso de uso

Como tiene acceso al lago de datos del parque de mascotas, ahora puede obtener datos para comenzar. Usando canalizaciones de datos (parte del paso de ingesta de datos), se hace lo siguiente:

1. Extraer, transformar y cargar 100000 imágenes de gatos y perros.
2. Dividir y versionar estos datos en una división de entrenamiento y prueba (con una división del 80 % y del 20 %).

El versionado de estos datos permitirá la trazabilidad de extremo a extremo para los modelos entrenados.

Ahora se está listo para comenzar a entrenar y probar el modelo de ML usando estos datos



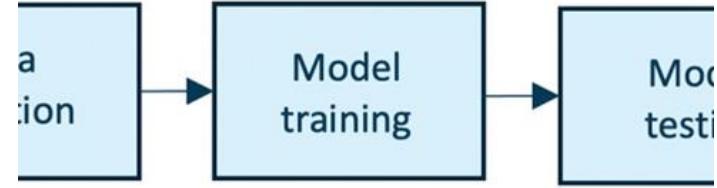
Entrenamiento del Modelo

- Después de obtener los datos necesarios para el entrenamiento de modelos de ML en el paso anterior, este paso habilitará el entrenamiento de modelos; tiene **scripts o códigos modulares** que realizan todos los pasos tradicionales en ML, como el **preprocesamiento de datos, la ingeniería de características** y el **escalado de características** antes de entrenar o volver a entrenar cualquier modelo.



Entrenamiento del Modelo

- Despues de esto, el modelo **ML se entrena** mientras se realiza el **ajuste de hiperparámetros** para ajustar el modelo al conjunto de datos (conjunto de entrenamiento). Este paso se puede realizar manualmente, pero existen soluciones eficientes y automáticas como **Grid Search** o Random Search.
- Como resultado, todos los pasos importantes del entrenamiento del modelo de ML se ejecutan con un **modelo de ML como resultado** de este paso.

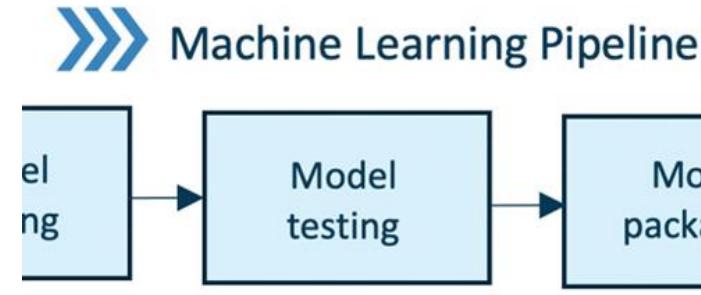


Entrenamiento del Modelo

Implementación de caso de uso

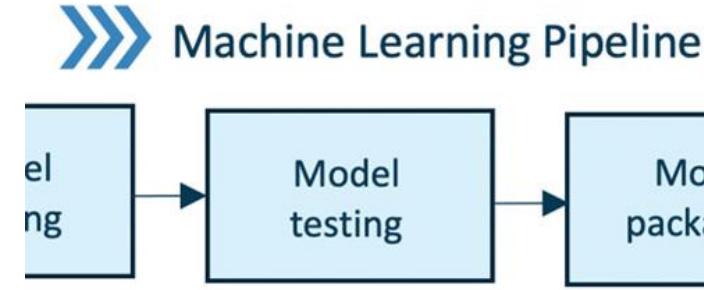
En este paso, implementamos todos los pasos importantes para entrenar el modelo de clasificación de imágenes. El objetivo es entrenar un modelo ML para clasificar gatos y perros. Para este caso, **entrenamos una red neuronal convolucional (CNN)** para el servicio de clasificación de imágenes. Se implementan los siguientes pasos: **preprocesamiento de datos, ingeniería de características y escalado de características antes del entrenamiento**, seguido de un entrenamiento del modelo con **ajuste de hiperparámetros**. Como resultado, tenemos un **modelo CNN para clasificar gatos y perros con un 97 % de precisión**.

Prueba del Modelo



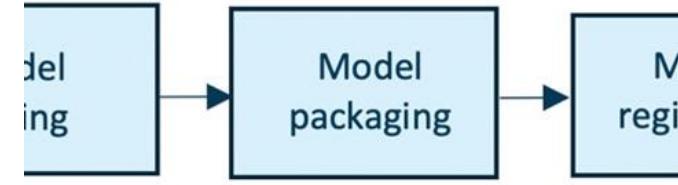
- En este paso, **evaluamos el rendimiento** del modelo entrenado en un conjunto separado de puntos de datos denominados **datos de prueba** (que se dividió y versionó en el paso de ingestión de datos). La inferencia del modelo entrenado se evalúa según **métricas seleccionadas** según el caso de uso. El **resultado** de este paso es un **informe sobre el rendimiento del modelo** entrenado.

Prueba del Modelo



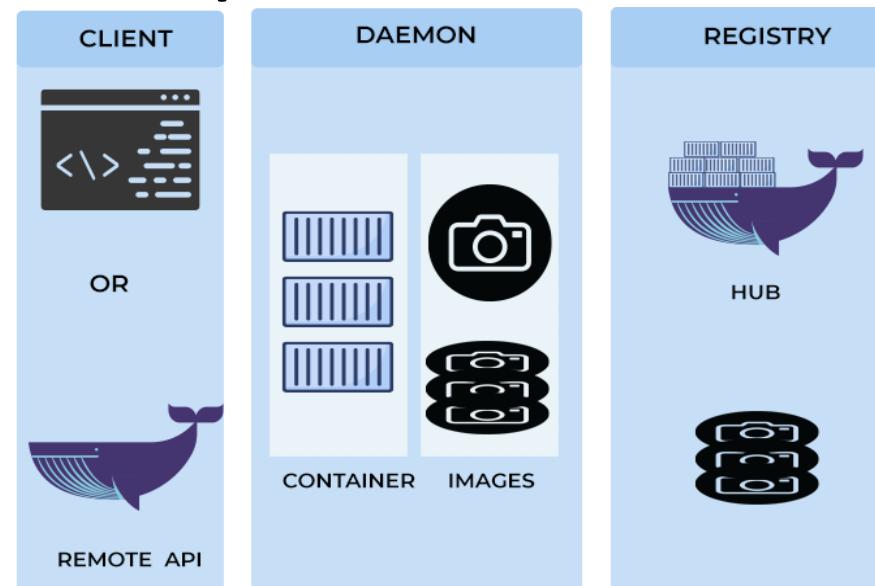
Implementación de casos de uso

Probamos el modelo entrenado en **datos de prueba** para **evaluar el rendimiento** del modelo entrenado. En este caso, buscamos **la precisión y sensibilidad** (“recall”) para **validar el rendimiento del modelo** en la clasificación de gatos y perros para evaluar los falsos positivos y los verdaderos positivos para obtener una comprensión realista del rendimiento del modelo. Si estamos satisfechos con los resultados y cuando estemos satisfechos con ellos, podemos continuar con el siguiente paso, o bien repetir los pasos anteriores para obtener un modelo de rendimiento decente para el servicio de clasificación de imágenes del parque de mascotas.



Empaquetado del modelo

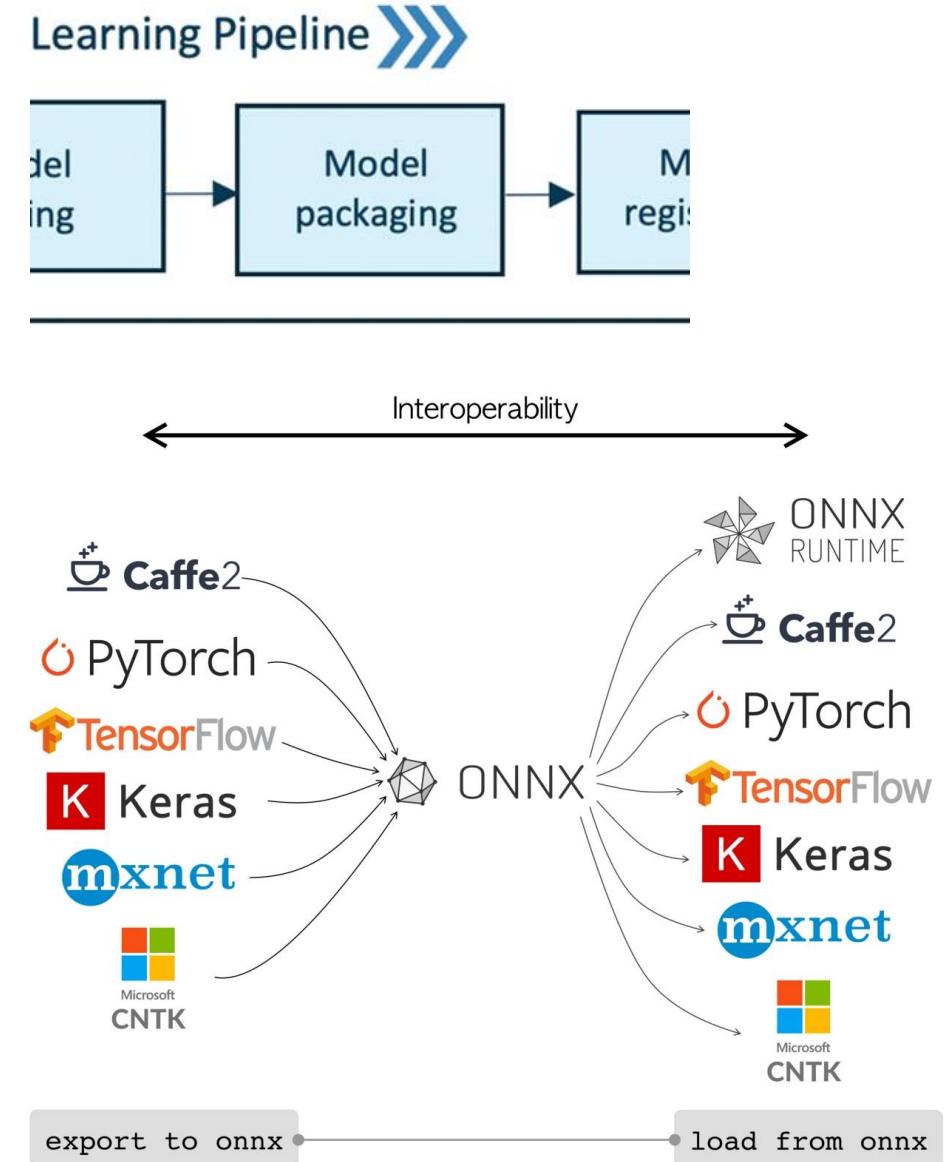
- Después de que el modelo entrenado se haya probado en el paso anterior, el modelo se puede serializar en un archivo o en un contenedor (usando Docker) para exportarlo al entorno de producción.



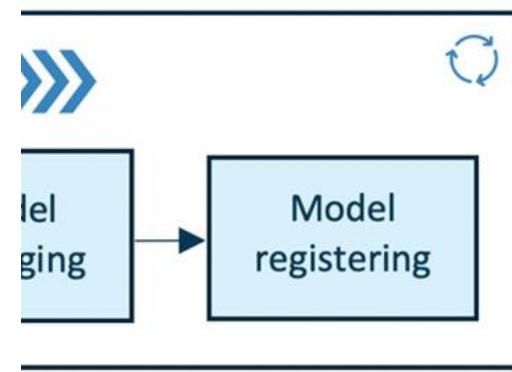
Empaque del modelo

Implementación del caso de uso

El modelo que entrenamos y probamos en los pasos anteriores es serializado en un archivo ONNX y está listo para desplegarse en el entorno de producción.

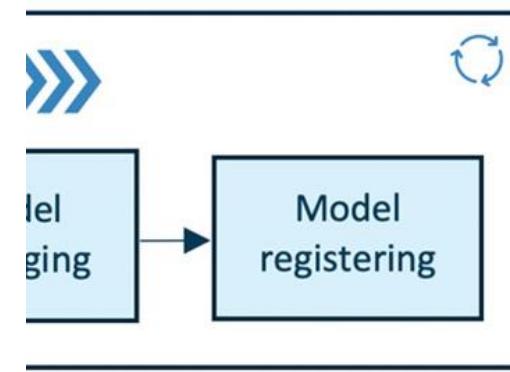


Registro del modelo



- En este paso, el **modelo que se serializó o contenerizo** en el paso anterior **se registra y almacena en el registro del modelo**. Un modelo registrado es una colección o paquete lógico de uno o más archivos que ensamblan, representan y ejecutan el modelo ML. Por ejemplo, varios archivos se pueden registrar como un modelo. Por ejemplo, un modelo de clasificación puede estar compuesto por un vectorizador, pesos de modelo y archivos de modelo serializados. Todos estos archivos se pueden registrar como un solo modelo. **Después de registrarse, el modelo** (todos los archivos o un solo archivo) **se puede descargar y desplegar según sea necesario**.

Registro del modelo



Implementación de caso de uso

El modelo serializado en el paso anterior se registra en el registro del modelo y está disponible para un despliegue rápido en el entorno de producción del parque de mascotas.

Finalmente

- Al implementar los pasos anteriores, ejecutamos con éxito la canalización de ML diseñada para nuestro caso de uso. Como resultado, tenemos modelos entrenados en el registro de modelos

