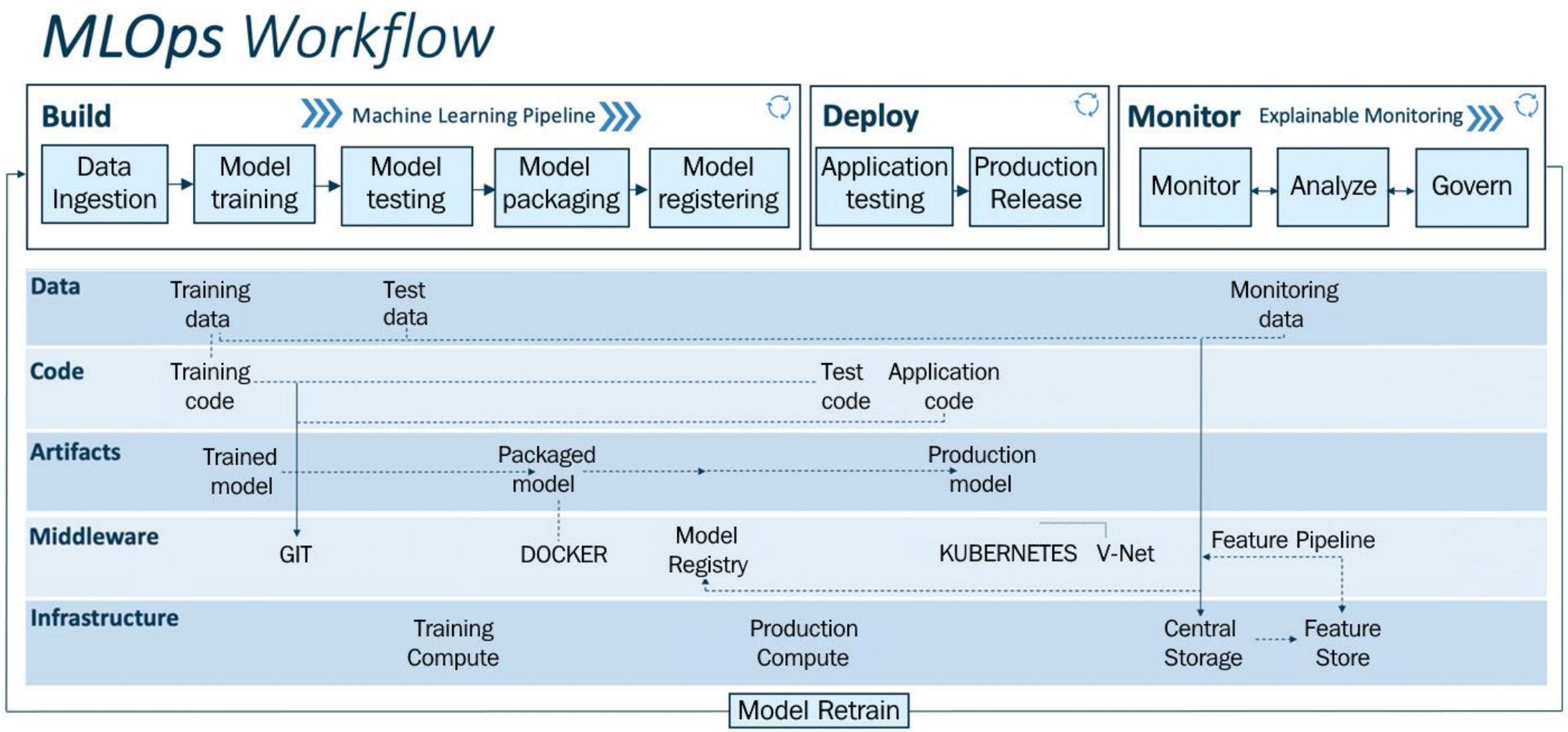
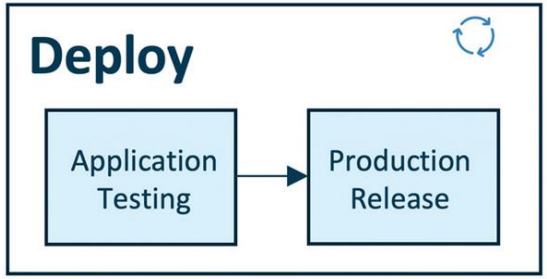
Conceptos y Flujo de trabajo de MLOps

2da Parte

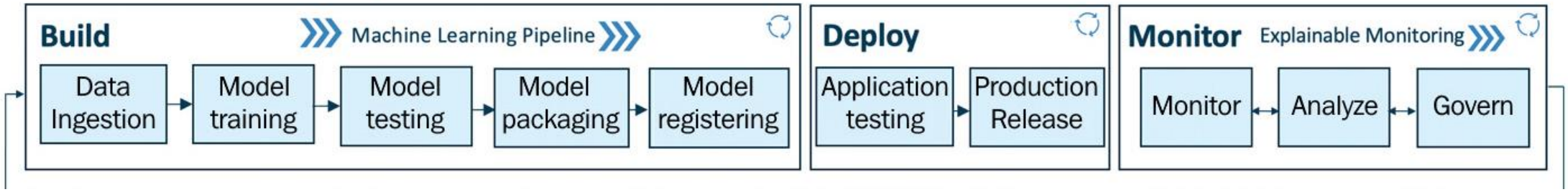
Flujo de Trabajo MLOps



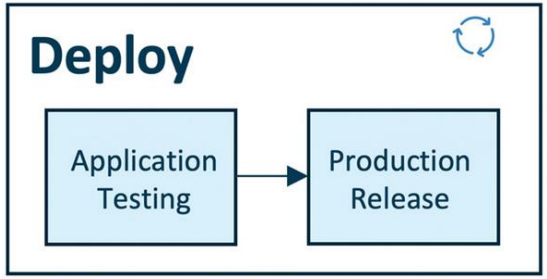


Despliegue

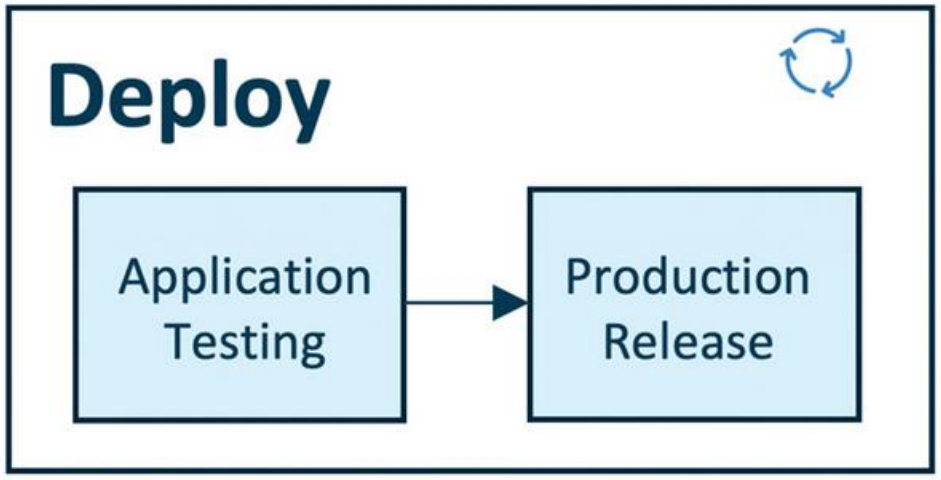
- El módulo de despliegue permite operacionalizar los modelos de ML que desarrollamos en el módulo anterior (construcción). En este módulo, probamos el rendimiento y el comportamiento de nuestro modelo en un entorno de producción o similar a la producción (prueba) para garantizar la solidez y la escalabilidad del modelo ML para uso en producción.

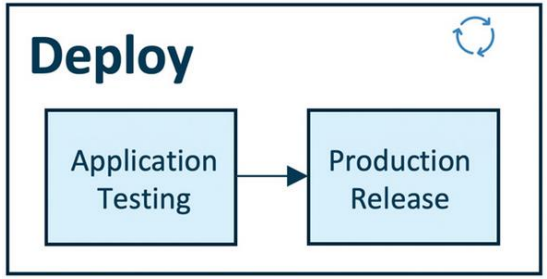


Despliegue



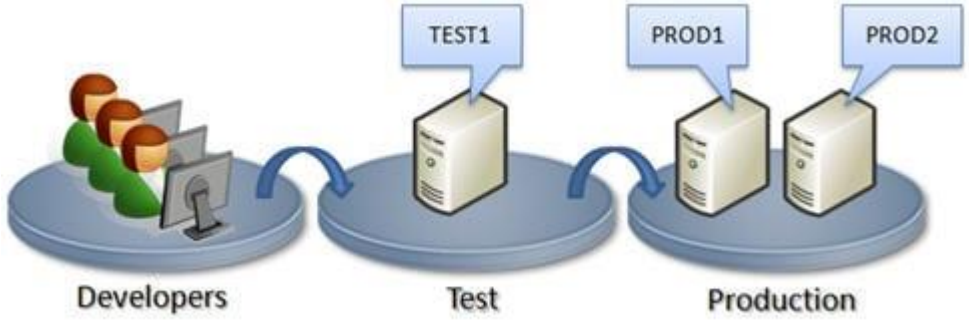
- La canalización del del despliegue tiene dos componentes (pruebas de producción y lanzamiento de producción) y está habilitada por canalizaciones de CI/CD optimizadas que conectan el desarrollo con los entornos de producción.



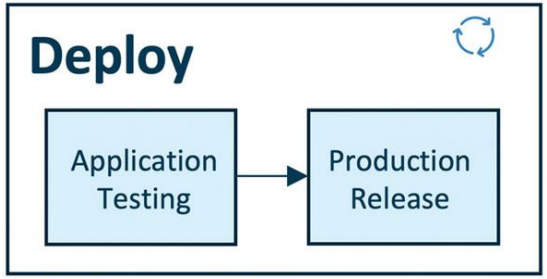


Prueba de la aplicación

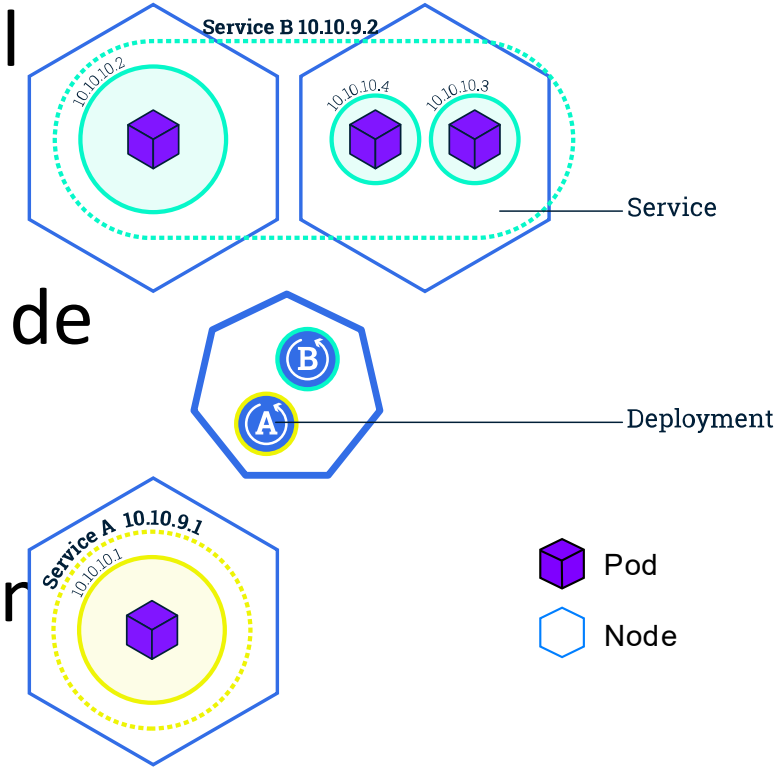
- Antes de implementar un modelo de ML en producción, es vital probar su solidez y rendimiento mediante pruebas. En la fase de "prueba de la aplicación" probamos rigurosamente todos los modelos entrenados en cuanto a robustez y rendimiento en un entorno similar al de producción llamado entorno de prueba. En la fase de prueba de la aplicación, desplegamos los modelos en el entorno de prueba (preproducción), que replica el entorno de producción.



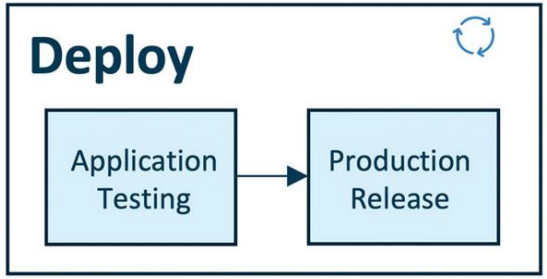
Prueba de la aplicación



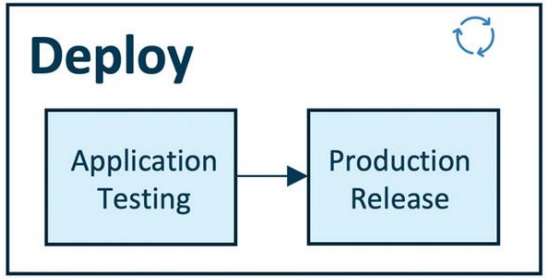
- El modelo de ML para pruebas se implementa como una API o servicio de transmisión en el entorno de prueba para objetivos de despliegue por ejemplo como clústeres de Kubernetes, instancias de contenedores o máquinas virtuales escalables o dispositivos de borde según la necesidad y el caso de uso. Después de desplegar el modelo para la prueba, se realizan predicciones utilizando datos de prueba (son datos de muestra de un entorno de producción) para el modelo desplegado para comprobar su robustes y rendimiento.



Prueba de la aplicación

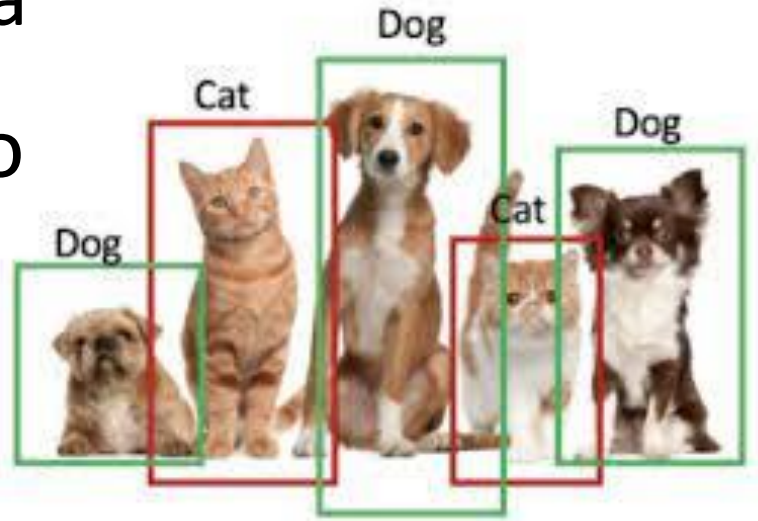


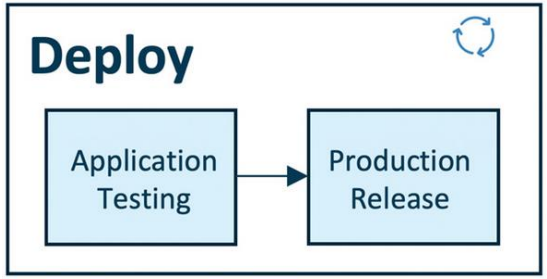
- Los resultados de rendimiento son revisados automática o manualmente por un experto en control de calidad. Cuando el rendimiento del modelo ML cumple con los estándares, se aprueba para implementarse en el entorno de producción donde el modelo se utilizará para inferir en lotes o en tiempo real para tomar decisiones comerciales.



Prueba de la aplicación

- **<u>Implementación de Caso de uso</u>**
- Implementamos el modelo como un servicio API en una computadora local en el parque de mascotas, que se configura con fines de prueba. Esta computadora está conectada a una cámara de circuito cerrado de televisión en el parque para obtener datos de inferencia en tiempo real para predecir gatos o perros en los cuadros de video.

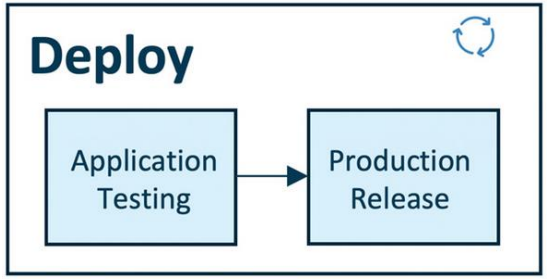




Prueba de la aplicación

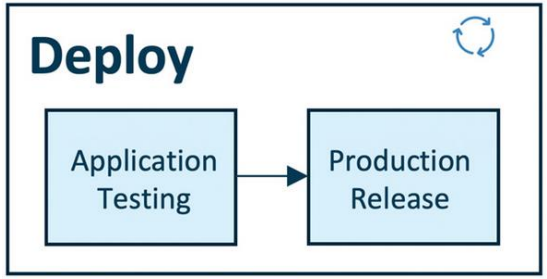
- **Implementación de Caso de uso**
- El despliegue del modelo está habilitada por la canalización de CI/CD. En este paso, probamos la robustez del modelo en un entorno similar de producción, es decir, si el modelo está realizando inferencias de forma consistente y un análisis de exactitud, equidad y errores. Al final de este paso, un experto en control de calidad certifica el modelo si cumple con los estándares.

Lanzamiento de Producción



- Los modelos previamente probados y aprobados se implementan en el entorno de producción para la inferencia del modelo para generar valor comercial u operativo. Esta versión de producción se implementa en el entorno de producción habilitado por canalizaciones de CI/CD.

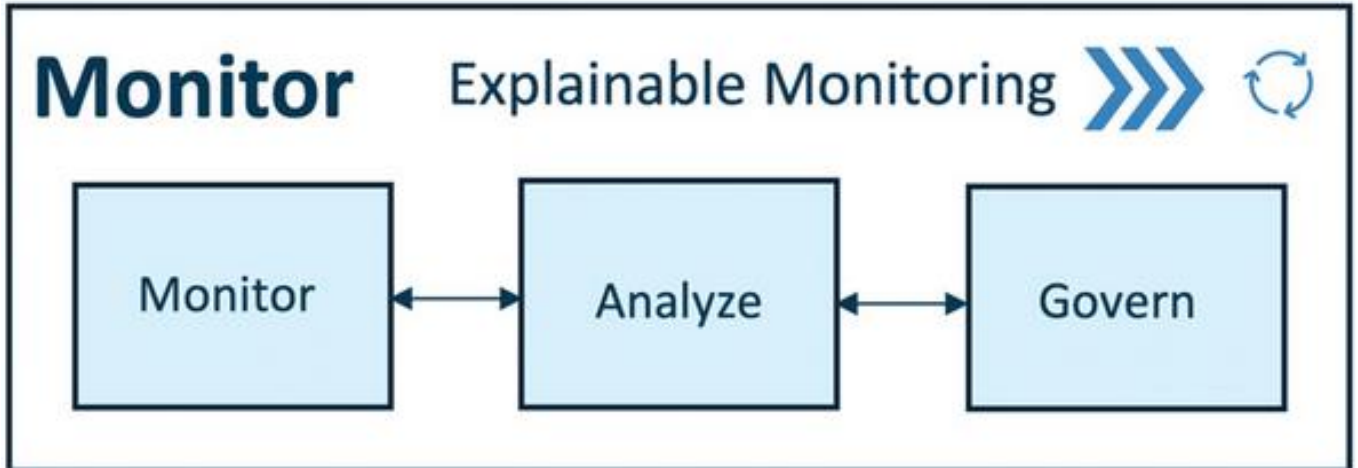
Lanzamiento de Producción



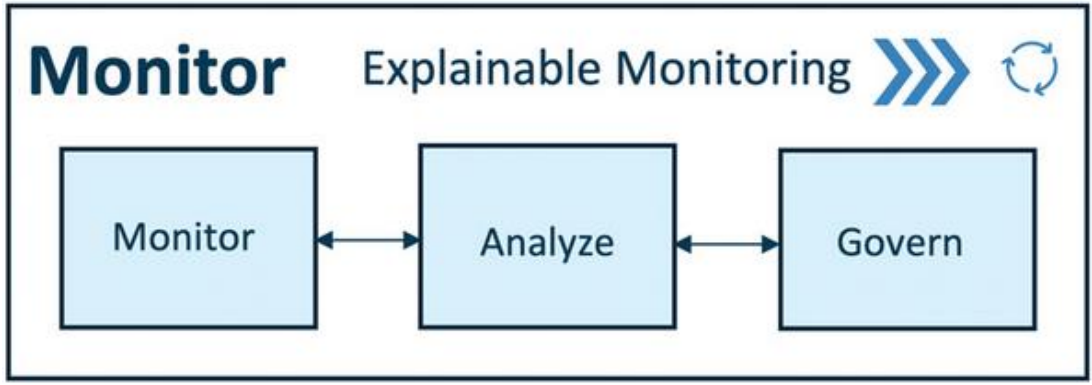
- **<u>Implementación de caso de uso</u>**
- Se despliega el modelo previamente probado y aprobado (por un experto en control de calidad) como un servicio API en una computadora conectada a CCTV en el parque de mascotas (configuración de producción). Este modelo implementado realiza una inferencia de ML en los datos de video entrantes de la cámara CCTV en el parque de mascotas para clasificar gatos o perros en tiempo real.

Monitorización

- El módulo de monitorización funciona sincronizado con el módulo de despliegue. Usando el monitoreo explicable, podemos monitorear, analizar y gobernar la aplicación ML implementada (modelo y aplicación ML).

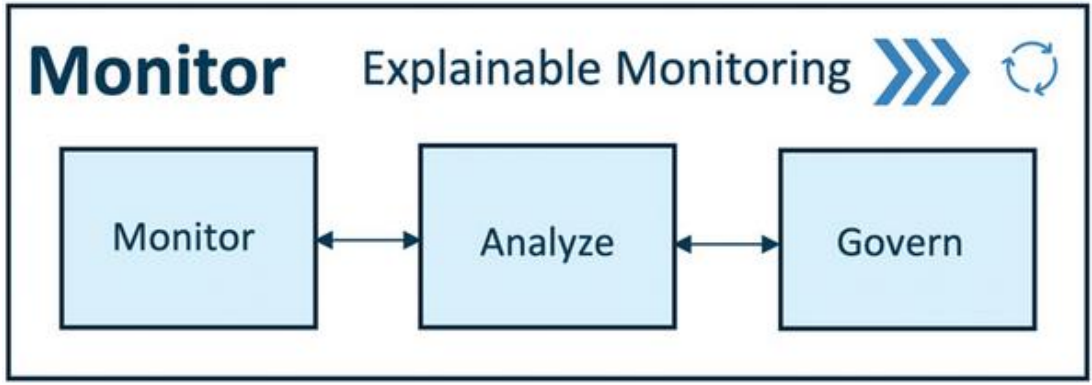


Monitorización



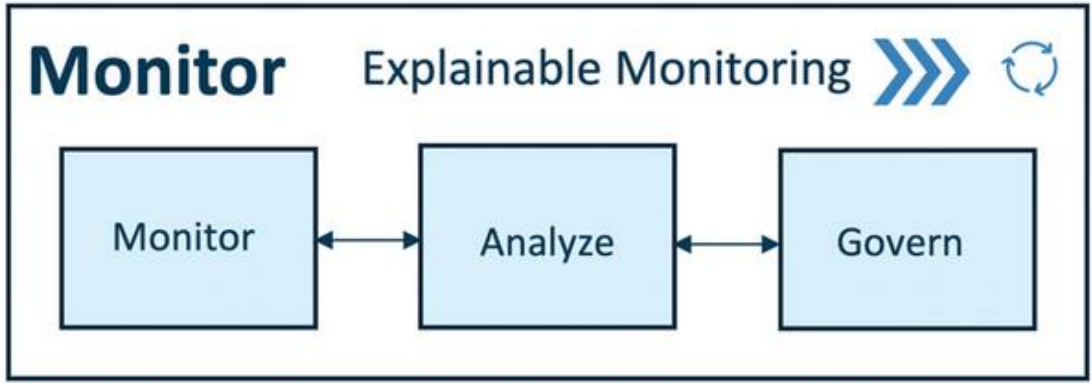
- En primer lugar, podemos monitorear el rendimiento del modelo ML (usando métricas predefinidas) y la aplicación implementada (usando datos de telemetría).
- En segundo lugar, el rendimiento del modelo se puede analizar utilizando un marco de explicabilidad predefinido
- Por último, la aplicación ML se puede gobernar mediante alertas y acciones basadas en el aseguramiento y control de calidad del modelo.

Monitorear



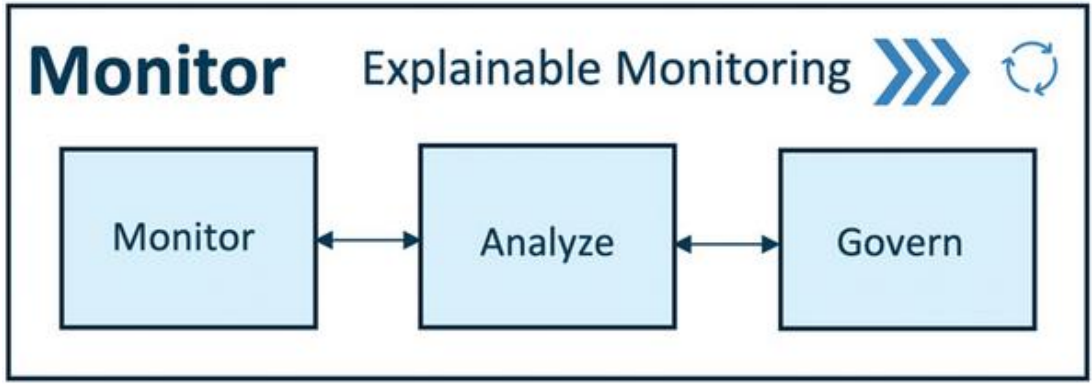
- El módulo de **monitoreo captura información crítica** para monitorear la integridad de los datos, la desviación del modelo y el rendimiento de la aplicación. El rendimiento de la aplicación se puede monitorear usando datos de telemetría. Representa el rendimiento del dispositivo de un sistema de producción durante un período de tiempo. Con datos de telemetría como de acelerómetro, giroscopio, humedad, magnetómetro, presión y temperatura, podemos controlar el rendimiento, estado y longevidad del sistema de producción.

Monitorear



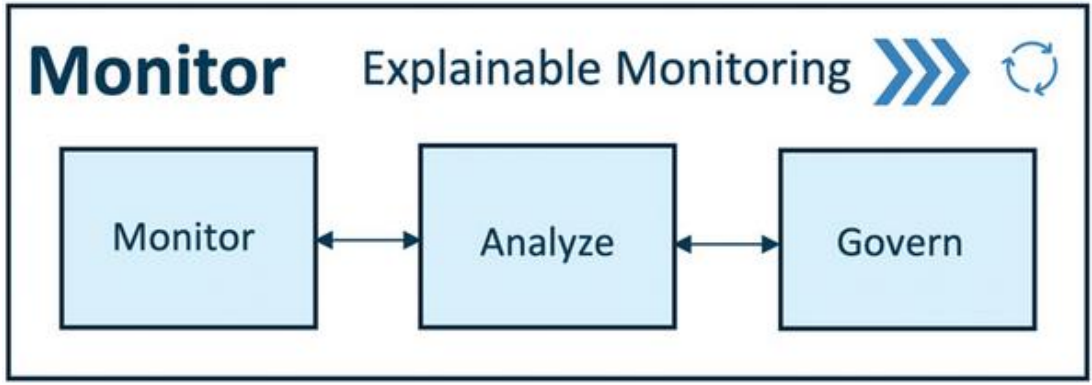
- **<u>Implementación de caso de uso</u>**
- En tiempo real, monitorearemos tres cosas: la integridad de los datos, la desviación del modelo y el rendimiento de la aplicación, para el servicio API implementado en la computadora del parque. Las métricas como la exactitud, la puntuación F1, la precisión y la recuperación se rastrean para la integridad de los datos y la desviación del modelo.

Monitorear



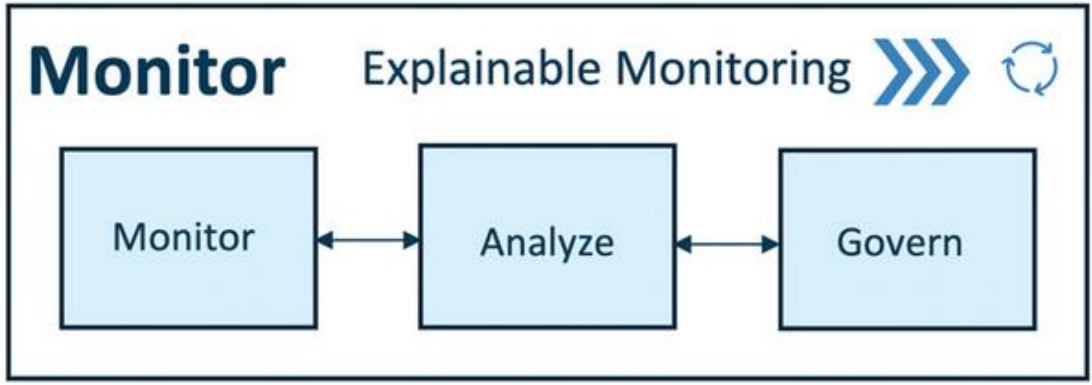
- **<u>Implementación de caso de uso</u>**
- Supervisamos el rendimiento de la aplicación mediante el seguimiento de los datos de telemetría del sistema de producción (la computadora local en el parque) que ejecuta el modelo ML implementado para garantizar el correcto funcionamiento del sistema de producción. Los datos de telemetría se monitorean para prever cualquier anomalía o falla potencial y corregirla con anticipación. Los datos de telemetría se registran y se pueden utilizar para evaluar el rendimiento del sistema de producción a lo largo del tiempo para comprobar su estado y longevidad.

Analizar



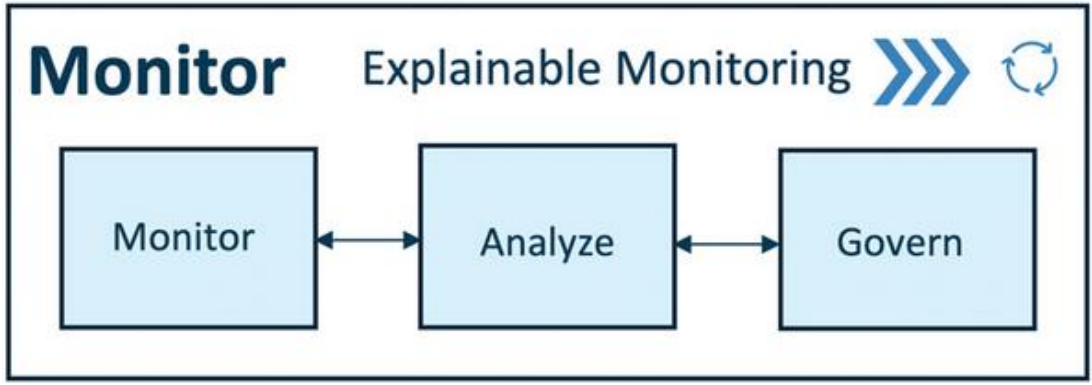
- Es fundamental **analizar el rendimiento** del modelo de los modelos de ML desplegados en los sistemas de producción para **garantizar un rendimiento y una gobernanza óptimos** en correlación con las decisiones o el impacto del negocio.
- Utilizamos **técnicas de explicabilidad** del modelo para medir el rendimiento del modelo en tiempo real. Con esto, evaluamos aspectos importantes como **la equidad del modelo, la confianza, el sesgo, la transparencia y el análisis de errores** con la intención de mejorar el modelo en correlación con el negocio.

Analizar



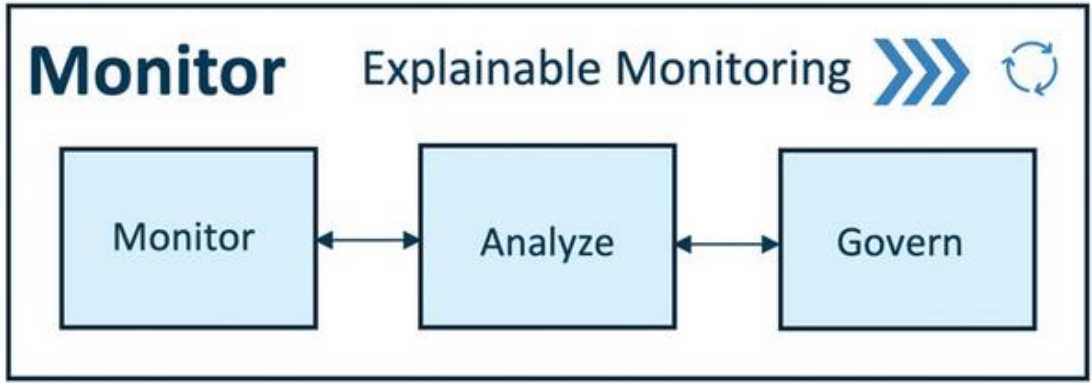
- Con el tiempo, **las propiedades estadísticas** de la variable **objetivo** que intentamos predecir **pueden cambiar** de forma imprevista. Este cambio se denomina "desviación del modelo", por ejemplo, en un caso en el que implementamos un modelo de sistema de recomendación para sugerir elementos adecuados para los usuarios. El comportamiento del usuario puede cambiar debido a tendencias imprevisibles que no se pudieron observar en los datos históricos que se usaron para entrenar el modelo. Es fundamental tener en cuenta estos factores imprevistos para garantizar que los modelos implementados proporcionen el mejor y más relevante valor de negocio.

Analizar



- Cuando se observa una desviación del modelo, se debe realizar cualquiera de estas acciones:
  a) Se debe alertar al propietario del producto o al experto de aseguramiento de la calidad.
  b) El modelo necesita ser cambiado o actualizado.
  c) Se debe activar el reentrenamiento de la canalización para volver a entrenar y actualizar el modelo según los últimos datos o necesidades

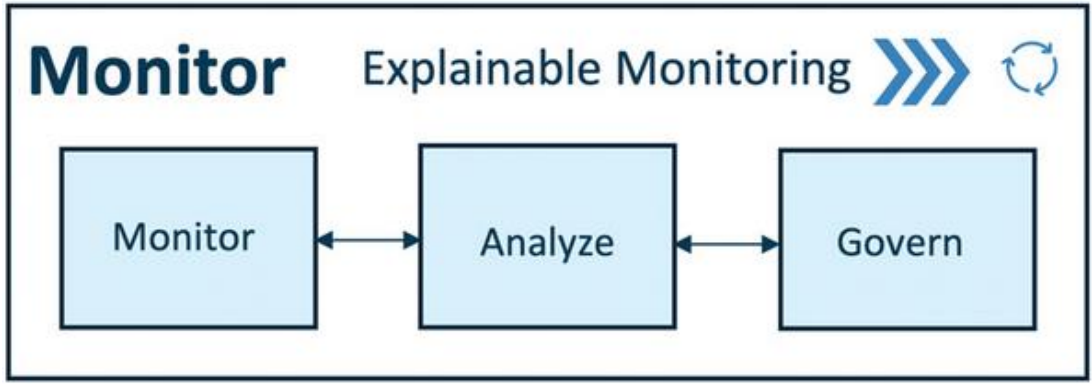
Analizar



Implementación de caso de uso

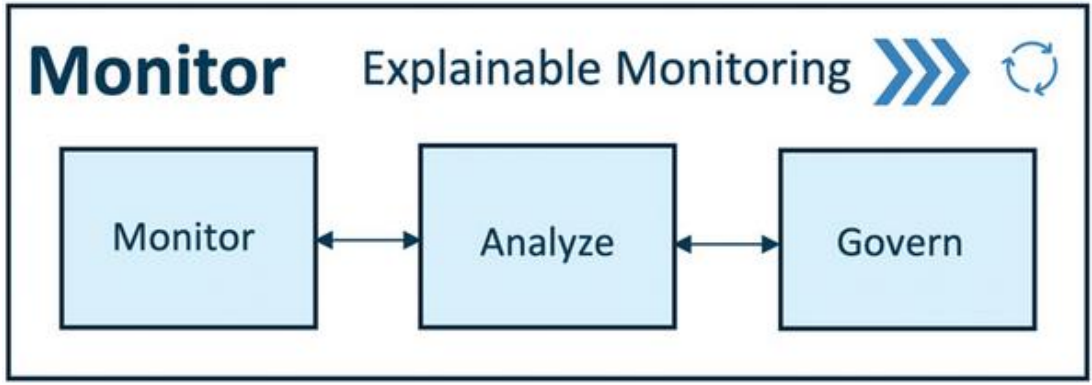
- Monitoreamos el rendimiento del modelo desplegado en el sistema de producción (un ordenador conectado al circuito cerrado de televisión del parque de mascotas). Analizaremos las puntuaciones de exactitud, precisión y recuperación del modelo periódicamente (una vez al día) para garantizar que el rendimiento del modelo no se deteriore por debajo del umbral. Cuando el rendimiento del modelo se deteriora por debajo del umbral, iniciamos mecanismos de control del sistema (por ejemplo, un disparador para volver a entrenar el modelo).

Gobernar



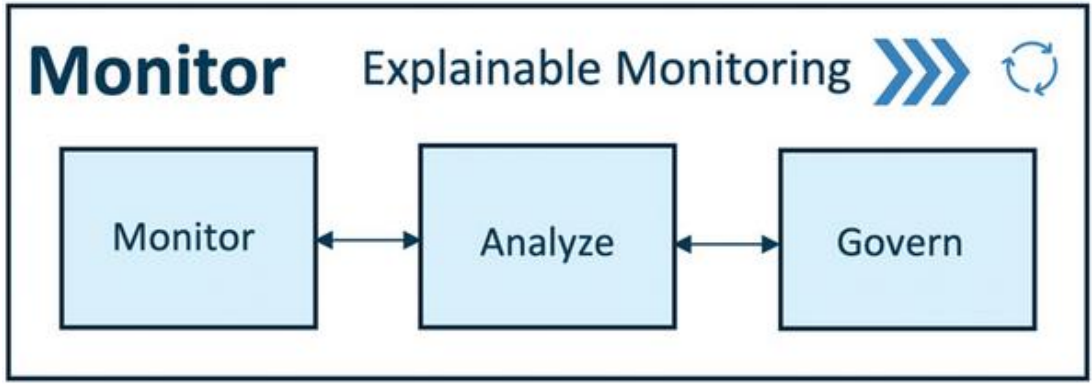
El monitoreo y el análisis se realizan para gobernar la aplicación implementada a fin de impulsar un rendimiento óptimo para el negocio (o el propósito del sistema ML). Luego de monitorear y analizar los datos de producción, podemos generar ciertas alertas y acciones para gobernar el sistema.

Gobernar



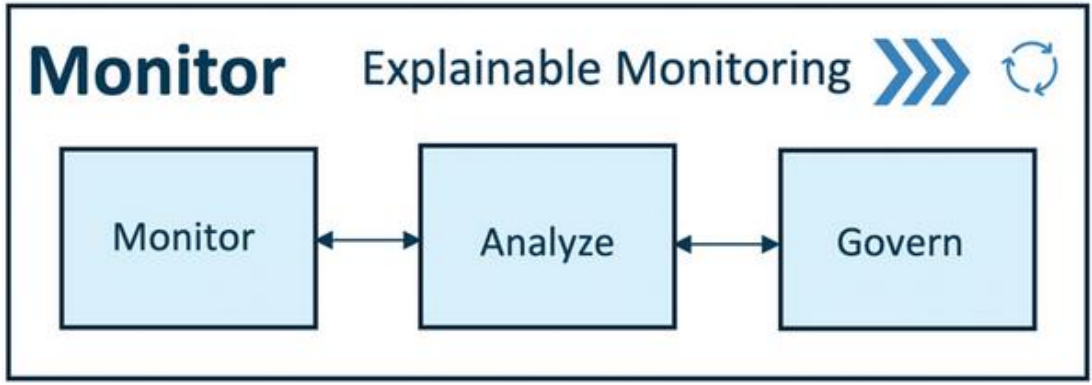
Por ejemplo, el propietario del producto o el experto en control de calidad recibe una alerta cuando el rendimiento del modelo se deteriora (por ejemplo, baja precisión, alto sesgo, etc.) por debajo de un umbral predefinido. El propietario del producto inicia un disparador para volver a entrenar e implementar un modelo alternativo.

Gobernar



Por último, un aspecto importante de la gobernanza es el "cumplimiento" de las leyes y normas locales y globales. Para el cumplimiento, la explicabilidad y la transparencia del modelo son vitales. Para esto, se realizan informes y auditorías de modelos para proporcionar trazabilidad y explicabilidad de extremo a extremo para los modelos de producción.

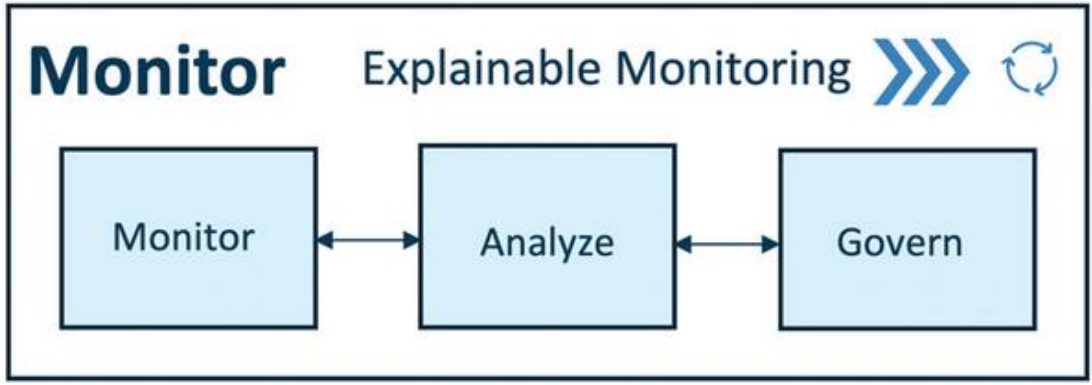
Gobernar



Implementación de casos de uso

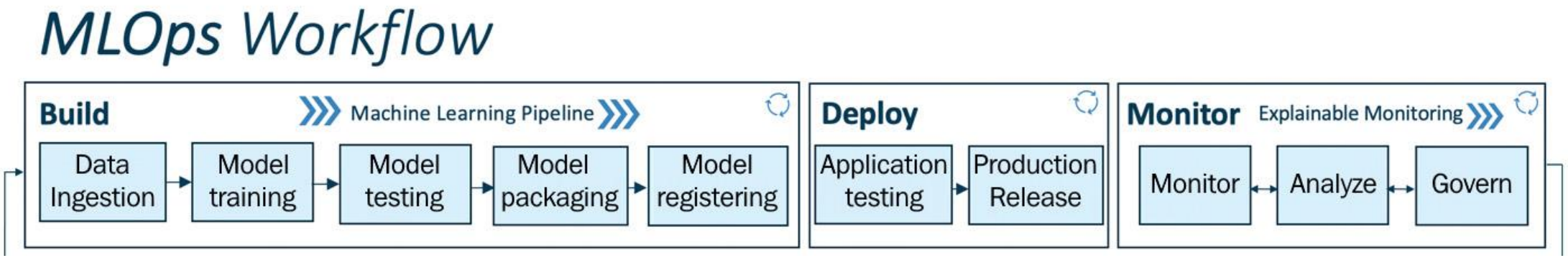
Monitorizamos y analizamos el rendimiento del modelo desplegado en el sistema de producción (un ordenador conectado al circuito cerrado de televisión del parque de mascotas). En función del análisis de las puntuaciones de exactitud, precisión y recuperación del modelo implementado, periódicamente (una vez al día) se generan alertas cuando el rendimiento del modelo se deteriora por debajo del umbral predefinido. El propietario del producto del parque genera acciones, y estas acciones se basan en las alertas.

Gobernar

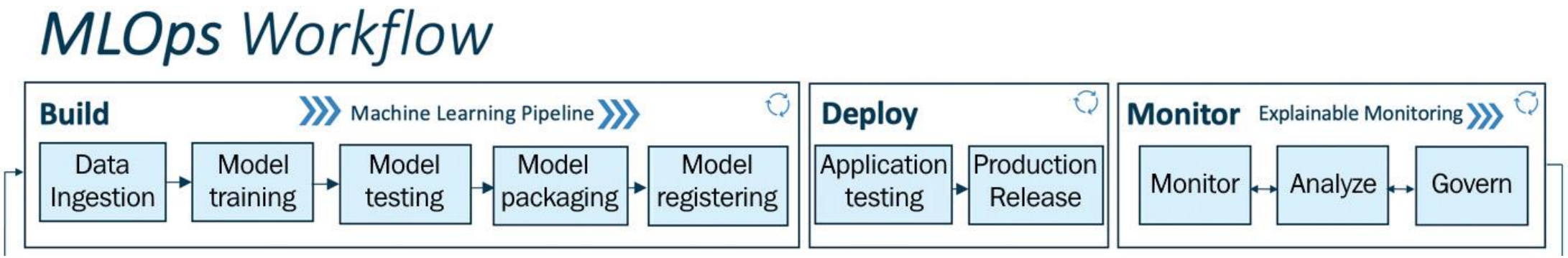


Implementación de casos de uso

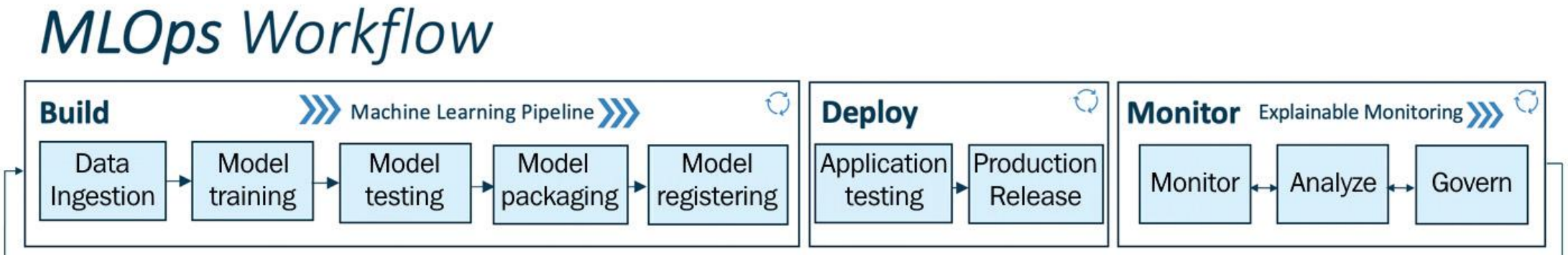
Por ejemplo, se genera una alerta notificando al dueño del producto que el modelo de producción está sesgado en un 30% para detectar perros más que gatos. Luego, el propietario del producto activa la canalización de reentrenamiento del modelo para actualizar el modelo utilizando los datos más recientes para reducir el sesgo, lo que da como resultado un modelo justo y robusto en producción. De esta manera, el sistema de ML en el parque de mascotas está bien gobernado para atender las necesidades del negocio.



- Todos los modelos entrenados, desplegados y monitoreados usando el método MLOps (Canalización MLOps) son rastreables de extremo a extremo y su versión se registra para rastrear los orígenes del modelo, que incluye el código fuente que el modelo usó para entrenar, los datos usados para entrenar y probar el modelo, y los parámetros utilizados para converger el modelo.

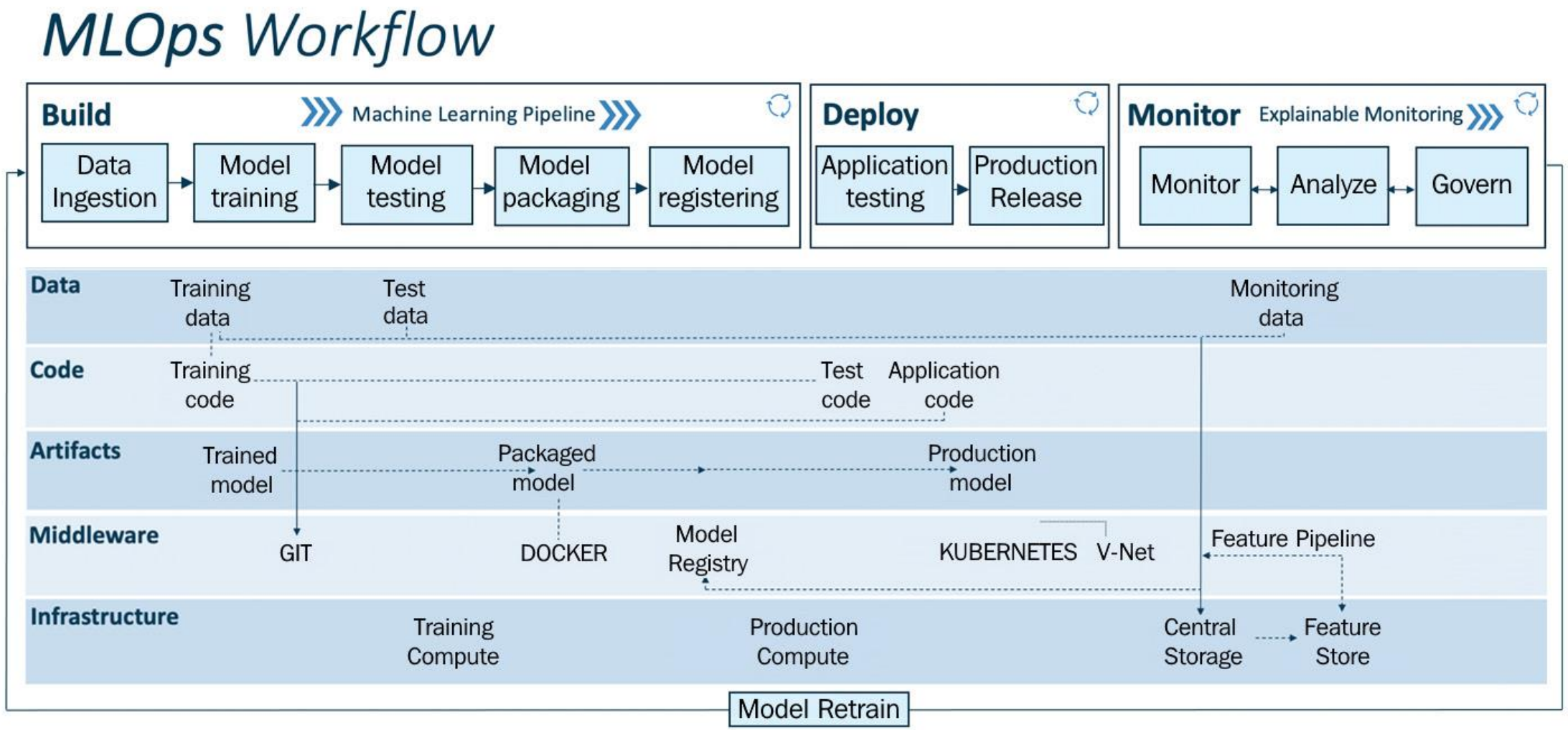


- La versión completa es útil para auditar operaciones o para replicar el modelo, o cuando falla, la versión del modelo de ML registrado es útil para rastrear los orígenes del modelo o para observar y depurar la causa del error.



- Como los modelos de ML generan datos en producción durante la inferencia, estos datos se pueden vincular a la versión de entrenamiento y despliegue del modelo para garantizar la versión de extremo a extremo, y esto es importante para ciertos requisitos de cumplimiento.
- A continuación, analizaremos los factores clave que permiten la canalización de MLOps.

Drivers


MLOps Workflow
Build
Machine Learning Pipeline
Data Ingestion
Model training
Model testing
Model packaging
Model registering
Deploy
Application testing
Production Release
Monitor
Explainable Monitoring
Monitor
Analyze
Govern
Data
Training data
Test data
Monitoring data
Code
Training code
Test code
Application code
Artifacts
Trained model
Packaged model
Production model
Middleware
GIT
DOCKER
Model Registry
KUBERNETES
V-Net
Feature Pipeline
Infrastructure
Training Compute
Production Compute
Central Storage
Feature Store
Model Retrain

Drivers

Datos: los datos pueden estar en múltiples formas, como texto, audio, video e imágenes. En las aplicaciones de software tradicionales, los datos suelen estar estructurados, mientras que, para las aplicaciones de ML, pueden estar estructurados o no estructurados. Para administrar datos en aplicaciones de ML, los datos se manejan en estos pasos: adquisición de datos, anotación de datos, catalogación de datos, preparación de datos, control de calidad de datos, muestreo de datos y aumento de datos. Cada paso implica su propio ciclo de vida. Esto hace que se necesite un conjunto completamente nuevo de procesos y herramientas para las aplicaciones de ML. Para un funcionamiento eficiente de la canalización de ML, los datos se segmentan y versionan en datos de entrenamiento, datos de prueba y datos de monitoreo (recopilados en producción, por ejemplo, entradas y salidas del modelo y datos de telemetría). Estas operaciones de datos son parte de la canalización de MLOps.

Drivers

Código: hay tres módulos esenciales de código que impulsan la canalización de MLOps: código de entrenamiento, código de prueba y código de aplicación. Estos scripts o códigos se ejecutan mediante CI/CD y canalizaciones de datos para garantizar el funcionamiento sólido de la canalización de MLOps. El sistema de administración de código fuente (por ejemplo, usando Git o Mercurial) permitirá la orquestación y jugará un papel vital en la administración e integración perfecta con CI, CD y canalizaciones de datos. Todo el código está preparado y versionado en la configuración de administración del código fuente (por ejemplo, Git).

Drivers

Artefactos: la canalización de MLOps genera artefactos como datos, modelos serializados, fragmentos de código, registros del sistema, entrenamiento de modelos de ML e información de métricas de prueba. Todos estos artefactos son útiles para el funcionamiento exitoso de la canalización de MLOps, lo que garantiza su trazabilidad y sostenibilidad. Estos artefactos se administran mediante servicios de middleware, como el registro de modelos, espacios de trabajo, servicios de registro, servicios de administración de código fuente, bases de datos, etc.

Drivers

Middleware: Middleware es un software informático que ofrece servicios a aplicaciones de software que son más de los que están disponibles en los sistemas operativos. Los servicios de middleware garantizan múltiples aplicaciones para automatizar y orquestar procesos para la canalización de MLOps. Podemos usar un conjunto diverso de software y servicios de middleware según los casos de uso, por ejemplo, Git para la administración del código fuente, VNets para habilitar las configuraciones de red requeridas, Docker para contener nuestros modelos y Kubernetes para la orquestación de contenedores para automatizar la implementación de aplicaciones. escalado y gestión.

Drivers

Infraestructura

Para garantizar el funcionamiento exitoso de la canalización de MLOps, necesitamos recursos informáticos y de almacenamiento esenciales para entrenar, probar e implementar los modelos de ML. Los recursos informáticos nos permiten entrenar, desplegar y monitorear nuestros modelos de ML.

Drivers

Infraestructura

Dos tipos de recursos de almacenamiento pueden facilitar las operaciones de ML, el almacenamiento central y el almacenamiento de características. Un almacenamiento central almacena los registros, los artefactos, la capacitación, las pruebas y los datos de monitoreo. Un almacén de características es opcional y complementaria al almacenamiento central. Extrae, transforma y almacena las características necesarias para el entrenamiento y la inferencia de modelos de ML mediante una canalización de características.

Drivers

Infraestructura

Cuando se trata de la infraestructura, existen varias opciones, como recursos locales o infraestructura como servicio (IaaS), que son servicios en la nube. En estos días, hay muchas opciones en la nube que brindan IaaS, como Amazon, Microsoft, Google, Alibaba, etc. Tener la infraestructura adecuada para su caso de uso permitirá operaciones sólidas, eficientes y fructiferas para su equipo y empresa.

Drivers

Infraestructura

Se puede lograr un flujo de trabajo totalmente automatizado con la optimización inteligente y la sinergia de todos estos drivers con la canalización de MLOps. Algunas ventajas directas de implementar un flujo de trabajo de MLOps automatizado son un aumento en la eficiencia de los equipos de TI (al reducir el tiempo que los desarrolladores y científicos de datos dedican a tareas mundanas y repetibles) y la optimización de los recursos, lo que resulta en reducciones de costos.