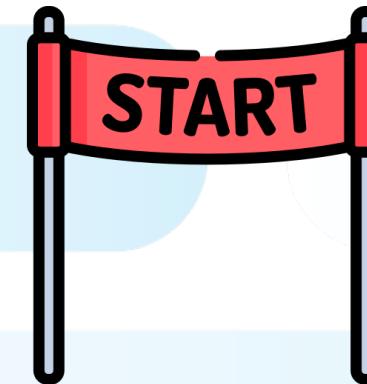


Tipos de despliegue



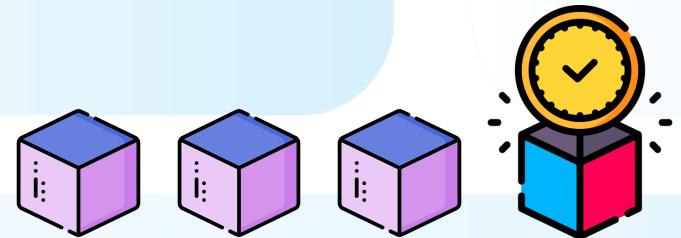
Tipos de despliegue

- ▶ Es una consideración que debes tener en mente desde el inicio del proyecto
- ▶ Se refieren a la forma y frecuencia con la que los modelos de machine learning van a generar predicciones



Batch

- ▶ Los modelos procesan datos en bloques a intervalos pre-determinados
- ▶ Las predicciones no son generadas inmediatamente
- ▶ Las predicciones son almacenadas para después ser consultadas



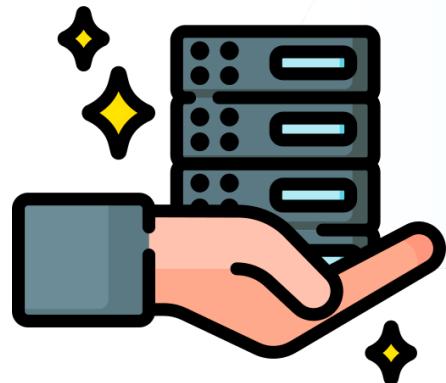
Ventajas de Batch

- **No hay muchas restricciones en el modelo:**
puedes usar modelos más complejos
optimizados para mejores resultados



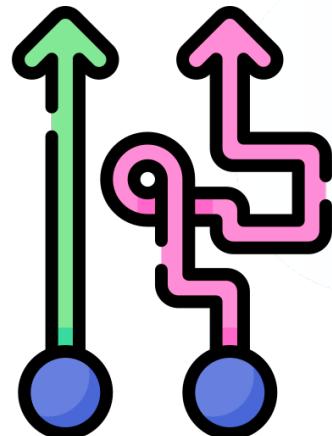
Ventajas de Batch

- **Mejor uso de recursos:** se aprovecha el máximo los recursos computacionales



Ventajas de Batch

- **Facilidad de implementación:** el procesamiento en modo batch es una práctica común



Ventajas de Batch

- Rapidez al obtener las predicciones: se deben consultar de una BD, más rápido que esperar a un modelo



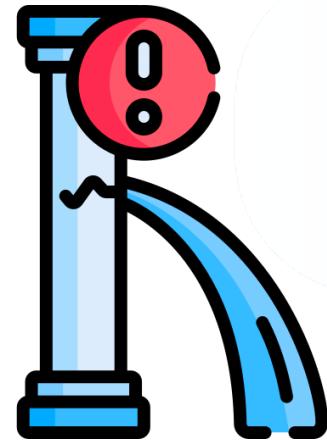
Desventajas de Batch

- **Lentitud al generar predicciones:** no es en tiempo real, está sujeto a una agenda de ejecución



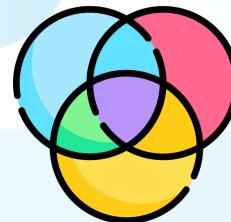
Desventajas de Batch

- **Consumo excesivo de recursos:** nos podemos dejar llevar y crear trabajos que consuman más tiempo del deseado.



Despliegue en Batch

- ▶ Extremadamente útil cuando el caso de uso se presta a esta solución
- ▶ Ten en cuenta las necesidades de tu cliente
- ▶ ¡Evita crear trabajos que se traslapen entre ellos!



Ejemplos de Batch

- ▶ Predecir una vez al día qué contenido va a ser el más visto en cada región



Ejemplos de Batch

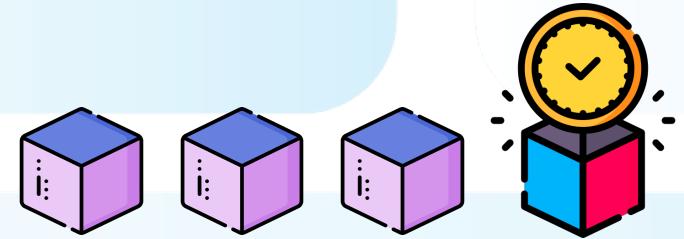
- ▶ Obtener las probabilidades de que un cliente cancele una reservación para llamarle y confirmar



Batch

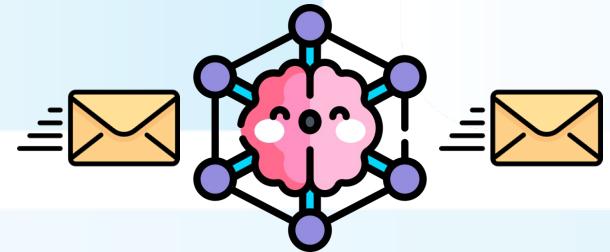
- ▶ Batch es conveniente y bien conocido
- ▶ Los modelos antes eran más pesados y tardaban más tiempo

pero...



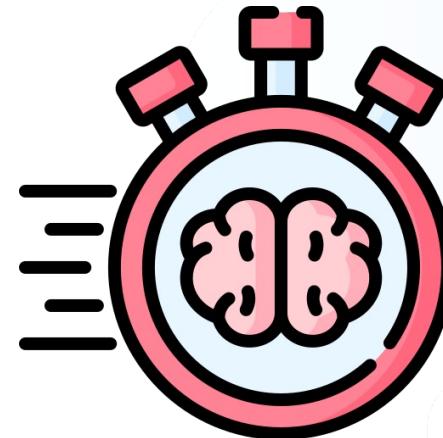
Online o en tiempo real

- ▶ Los modelos procesan observaciones en cuanto estas llegan al servidor
- ▶ Actúa sobre una sola observación a la vez
- ▶ Se puede integrar con sistemas interactivos



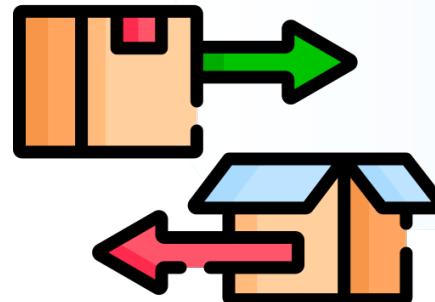
Ventajas de *online*

- **Velocidad y baja latencia:** se ofrecen respuestas casi instantáneas, el modelo está optimizado para este caso de uso



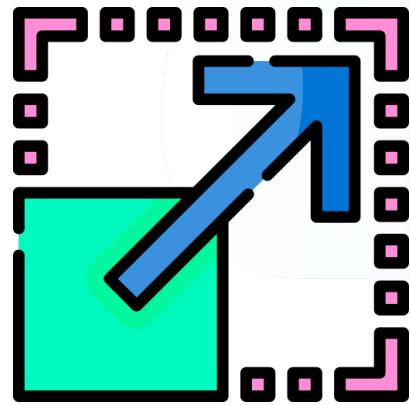
Ventajas de *online*

- **Actualización rápida de modelos:** permite intercambiar modelos rápidamente y de manera silenciosa



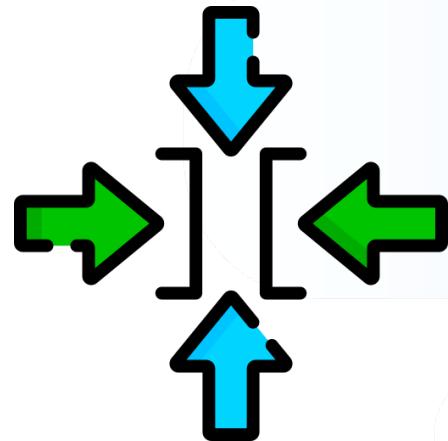
Ventajas de *online*

- **Mayor capacidad de escalamiento:** permite escalar de forma rápida en caso de que se requiera mayor volumen de inferencia



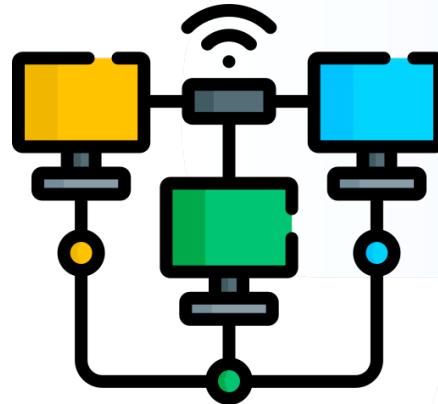
Desventajas de *online*

- **Restricciones en el modelo:** suelen ser de una complejidad menor y capacidad marginalmente reducida



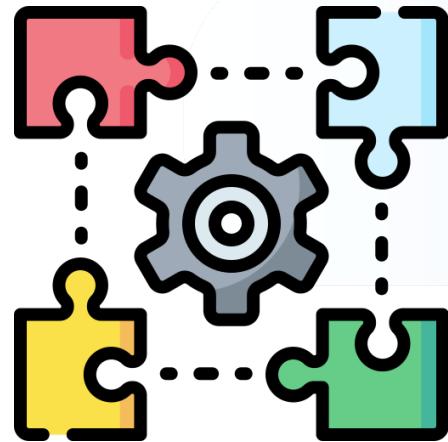
Desventajas de *online*

- **Dependencia de factores externos:** la velocidad (y utilidad del modelo) depende de otros aspectos, como la red



Desventajas de *online*

- **Complejidad de despliegue:** requiere de muchos más componentes que ayudan a que el modelo funcione tan rápido como se espera



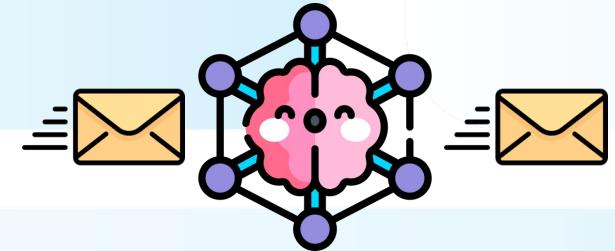
Desventajas de *online*

- **Costo de la infraestructura:** dependiendo de la complejidad del despliegue, el costo es otro aspecto a considerar



Online o en tiempo real

- ▶ Es comprensible querer que todo funcione on-line
- ▶ Aún así es crucial identificar la necesidad real de nuestros clientes



Ejemplos *Online*

- ▶ Un chatbot que esté disponible 24/7 para los usuarios



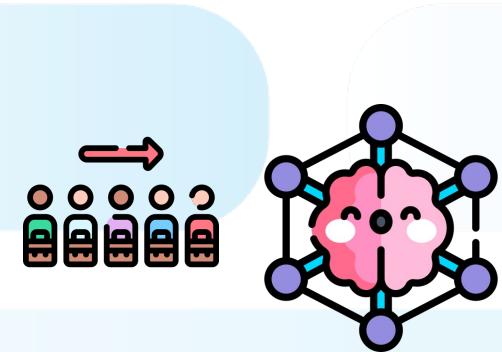
Ejemplos *Online*

- ▶ Un sistema de recomendación de actividades para huéspedes de un hotel



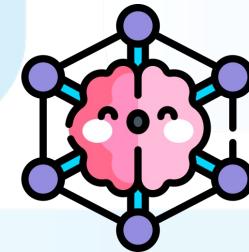
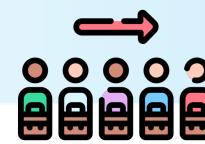
Streaming

- ▶ Se ubica entre el modo *batch* y *online*
- ▶ Los datos son procesados tan pronto como sea posible pero no en tiempo real
- ▶ Opera constantemente sobre un flujo de datos



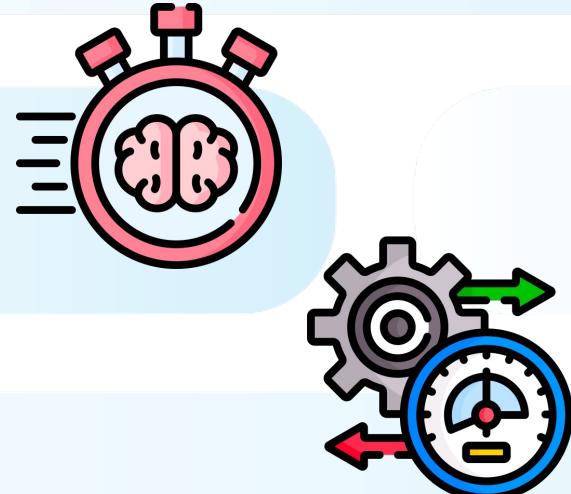
Streaming

- ▶ Se usa cuando no se espera una respuesta inmediata pero tampoco se desea esperar a un intervalo fijo
- ▶ Cuando la latencia es importante pero no crítica



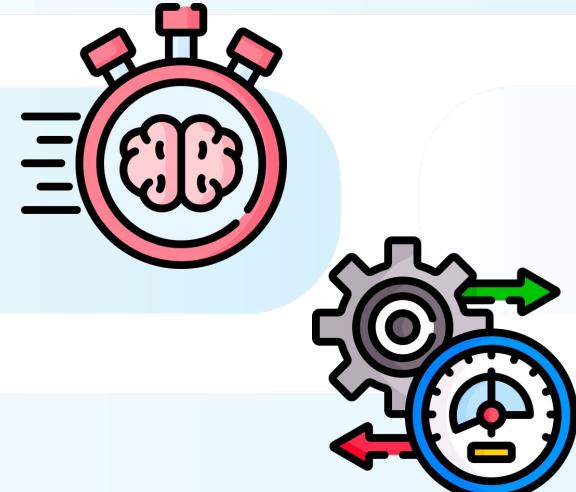
Ventajas *Streaming*

- **Latencia reducida:** más rápido que batch
- **Mayor flexibilidad al crear un modelo:** no existen tantas restricciones al crear un modelo



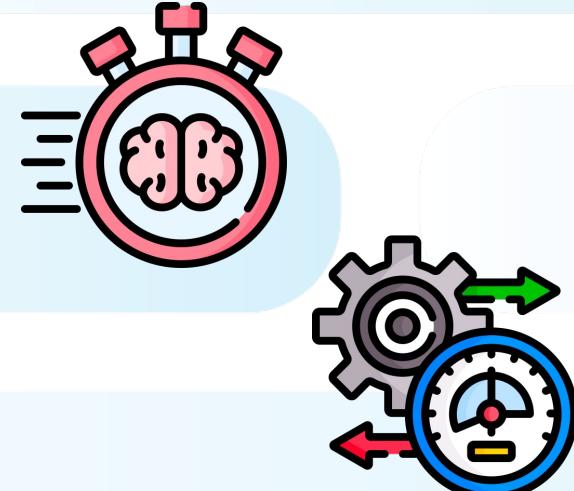
Desventajas *Streaming*

- **Mayor complejidad de implementación:** en un sistema en *streaming* hay otras piezas que hay que considerar
- **Complejidad al escalar:** requiere de más configuración debido a la necesidad de manejar flujos de datos



Ejemplos *Streaming*

- **Mayor complejidad de implementación:** en un sistema en *streaming* hay otras piezas que hay que considerar
- **Complejidad al escalar:** requiere de más configuración debido a la necesidad de manejar flujos de datos



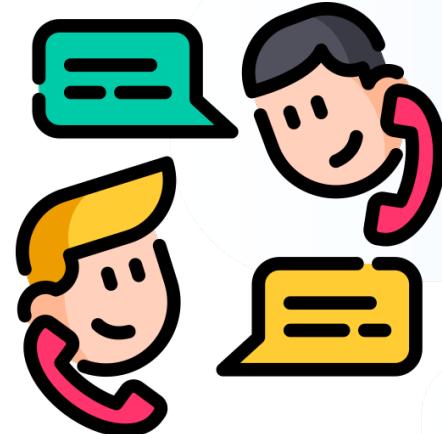
Ejemplos *Streaming*

- ▶ Un modelo que detecta fraudes financieros: es importante que sucedan rápido, pero no en tiempo real



Ejemplos *Streaming*

- ▶ Un modelo que calcula la probabilidad de convertir un cliente si se le llama después de que abandonó el formulario



Tipos de despliegue

- Batch, streaming, online
- Es importante tener en consideración el problema que vamos a resolver antes de tomar una decisión de qué tipo vamos a utilizar

