# Propuesta para despliegue en producción de los modelos

1. Análisis del caso.

Uno de los retos técnicos que se puede identificar es que las imágenes pueden variar considerablemente en términos de resolución, tamaño, contraste e incluso formato (jpg, png o pdf). Igualmente, el modelo fue entrenado con notas clínicas en inglés lo que lo limita a este idioma. Asimismo, las notas clínicas podrían contener abreviaturas o errores de digitación.

Una forma de solucionar lo anteriormente planteado, es implementando una capa de ingesta y preprocesamiento muy robusta que normalice los formatos y la valide que sean los adecuados, como por ejemplo que la imagen abra correctamente y que el texto no se encuentre vacío.

Con el tiempo, se debe poder reentrenar los modelos con los datos recopilados a la fecha. Para esto, un pipeline de MLOps que automatice versionado y validación podría ser útil.

Sumado a esto, el riesgo de que, dado que se trabajó con datos de salud, estos son extremadamente sensibles por lo que deberían estar almacenados en un lugar seguro.

Por último, otro riesgo de ambos modelos es que no permite brindar explicabilidad clínica a los médicos lo que podría desencadenar una baja adopción por parte de estos.

1. Diseño de la solución.
   1. Ingesta: Datos batch, usar un Blob Storage para almacenar los datos. Tanto las imágenes como las notas clínicas. Lo ideal crear un contenedor raw para cada uno.
   2. Preprocesamiento: Usar Azure Data Factory para orquestar los flujos de trabajo para mover los archivos de raw a un área de procesamientos. Después, con Azure functions realizar los preprocesamientos de los datos como el “CropBlackBorders” en el caso de las imágenes.
   3. Azure Data Lake para almacenar los datos procesados.
   4. Azure Machine Learning Workspace nos permite orquestar los entrenamientos de ambos modelos. Acá se puede crear Pipelines de MLOps usando como trigger cada vez que se sube nueva data al Blob Storage.
   5. Deployment: Se configura un Azure ML Endpoint teniendo los modelos finales. Con Azure Kubernetes Service para crear un cluster con nodos GPU y publicar un servicio web usando contenedores de Docker para cada uno de los modelos.
2. Evaluación y gobernanza de la solución.

Para la evaluación de los modelos, se pueden implementar métricas como data drift como cambios fuertes en la distribución de las predicciones.

También, realizar estudios retrospectivos y prospectivos donde se comparen las predicciones vs el diagnostico y tratamiento real propuesto por el médico.