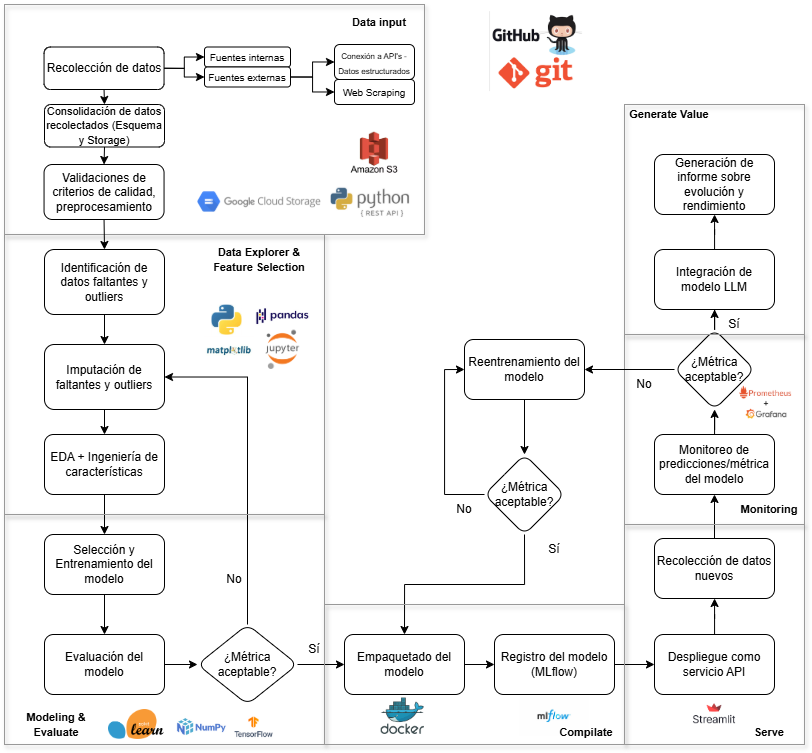
**Diseño**

1. **Diseño del pipeline de MLOps**



1. **Etapas del pipeline**
   1. **Recolección de datos**. En esta etapa se recopila información clínica de los pacientes a partir de fuentes internas como historias clínicas electrónicas y registros hospitalarios, así como de fuentes externas (las disponibles) como APIs médicas y web scraping. Esta información puede incluir síntomas, diagnósticos previos, tratamientos y resultados de laboratorio.
   2. **Validaciones de criterios de calidad, preprocesamiento.** En esta etapa se revisan y contrastan los datos recolectados con criterios de calidad previamente definidos por el equipo para asegurar su integridad, coherencia y formato adecuado. Además, se eliminan errores, duplicados o inconsistencias que puedan afectar negativamente el entrenamiento del modelo.
   3. **Identificación de datos faltantes y outliers.** En esta etapa se analizan los datos para detectar valores ausentes o atípicos (outliers), que pueden comprometer la calidad del modelo
   4. **Imputación de faltantes y outliers.** En esta etapa se aplican técnicas para completar datos faltantes (como imputación por media o modelos multivariados) y tratar valores extremos, permitiendo conservar el mayor número de registros posible sin afectar la validez del análisis.
   5. **Ingeniería de características.** En esta etapa se transforman y crean variables a partir de los datos originales, como por ejemplo convertir síntomas escritos en texto en variables numéricas o combinar características clínicas relevantes
   6. **Selección y entrenamiento del modelo.** En esta etapa se eligen y entrenan algoritmos de aprendizaje supervisado adecuados para el tipo de problema (clasificación multiclase), permitiendo al modelo aprender a predecir la posible presencia de una enfermedad a partir de los síntomas.
   7. **Evaluación del modelo.** En esta etapa se evalúa el rendimiento del modelo entrenado utilizando métricas como F1-score, AUC, precisión o recall, comparando su capacidad predictiva para enfermedades
   8. **¿Métrica aceptable? (condicional).** En esta etapa se toma una decisión sobre si el desempeño del modelo es suficiente según métricas clínicas definidas. Si no lo es, se ajusta el flujo anterior (ingeniería, imputación o entrenamiento).
   9. **Empaquetado del modelo.** En esta etapa se agrupa el modelo final y sus dependencias en un entorno portátil (usando Docker), dejándolo listo para ser desplegado en producción de forma consistente y reproducible.
   10. **Registro del modelo (MLflow).** En esta etapa se almacena el modelo, junto con su versión, hiperparámetros, métricas de desempeño y contexto de entrenamiento, en una herramienta como MLflow, para facilitar su auditoria y trazabilidad
   11. **Despliegue como servicio API.** En esta etapa se expone el modelo como un servicio web accesible mediante API, permitiendo que otras personas lo consulten para realizar predicciones de manera automática y en tiempo real (consumo del modelo).
   12. **Recolección de datos nuevos.** En esta etapa se recogen nuevos registros de pacientes, ya sea provenientes del uso real del sistema, lo cual alimenta el proceso de actualización del modelo.
   13. **Monitoreo de predicciones/métrica del modelo.** En esta etapa se supervisa el desempeño del modelo una vez en producción, observando si sus predicciones mantienen la calidad esperada o si hay señales de degradación (data drift)
   14. **¿Métrica aceptable? (condicional).** En esta etapa se decide si el modelo sigue siendo confiable con los nuevos datos. Si no lo es, se activa el proceso de reentrenamiento para actualizar su conocimiento.
   15. **Reentrenamiento del modelo.** En esta etapa se reutilizan los nuevos datos recopilados para volver a entrenar el modelo, ajustarlo a nuevas condiciones o patrones emergentes, y mejorar su precisión.
   16. **¿Métrica aceptable? (condicional).** En esta etapa se evalúa si el modelo reentrenado ha mejorado su rendimiento y cumple con los requisitos clínicos. Si no lo hace, se itera nuevamente; si lo logra, se continúa al despliegue.
   17. **Integración de modelo LLM.** En esta etapa se incorpora un modelo de lenguaje natural (LLM) que sirva para la etapa de generación de informes y pueda generar interpretaciones, resúmenes o explicaciones sobre las predicciones del modelo de ML, facilitando la comprensión para médicos o pacientes (amigable de cara al usuario).
   18. **Generación de informe sobre rendimiento y evolución del modelo.** En esta etapa se elaboran reportes automáticos sobre la evolución del modelo, sus métricas a lo largo del tiempo y las decisiones tomadas, lo cual aporta transparencia y respaldo para decisiones clínicas. (amigable para el equipo de datos)
2. **Restricciones, limitaciones y tipos de datos**

|  |  |  |  |
| --- | --- | --- | --- |
| **Etapa** | **Restricciones** | **Limitaciones** | **Tipos de datos** |
| **Recolección de datos**. | Privacidad de los datos, restricciones legales, burocracia institucional | Poca disponibilidad de datos de enfermedades huérfanas, bajo nivel de digitalización de las entidades, | Datos estructurados, no estructurados |
| **Validaciones de criterios de calidad, preprocesamiento** | Calidad heterogénea provenientes de varias fuentes | Lograr consensos con entidades sobre el significado de un dato, presencia de errores, formatos inconsistentes, ambigüedades | Variables numéricas, categóricas, texto libre. |
| **Identificación e imputación de faltantes, outliers** | Restricciones de imputación por sensibilidad en el dato, sesgos | Imputación de faltantes o outliers sin contextos clínicos completos, |  |
| **Entrenamiento del modelo** | Desequilibro de las clases (enfermedades normales y huérfanas) | Insuficiencia de datos etiquetados para diagnóstico de enfermedades huérfanas |  |
| **Evaluación del modelo** | Métricas específicas por enfermedad normal y enfermedad huérfana, buen rendimiento del modelo para una de las dos clases | Métrica genérica como el accuracy no es la más conveniente, |  |
| **Empaquetado** | Si el modelo se embebe en otro servicio la compatibilidad puede ser restrictiva, regulaciones sobre software medico si es diagnóstico asistido |  | .pkl |
| **Registro del modelo** | Regulatorias al almacenar datos sensibles | Para escalar el modelo pueden haber dificultades si no se cuenta con infraestructura solida | Metadatos del entorno, versiones |
| **Despliegue como servicio API** | Validaciones clínicas previas, cumplimiento de normatividad en uso de datos y sector salud | Dependencia de infraestructura tech hospitalaria para integrar a sistemas reales |  |
| **Recolección de datos nuevos** | Garantizar que los nuevos datos sean anonimizados | Frecuencia de datos nuevos |  |
| **Reentrenamiento del modelo** | Validaciones médicas son obligatorias | Pocos datos nuevos para el reentrenamiento del modelo, costos computacionales |  |
| **Integración de modelo LLM (como apoyo, NO COMO DIAGNÓSTICO)** | LLMs deben evitar generar información médica incorrecta o no verificada (“alucinaciones”). Debe haber trazabilidad de lo generado por el modelo LLM | Alto consumo de recursos (memoria, procesamiento). | Informes, texto. |
| **Generación de informe sobre rendimiento y evolución del modelo** | Los reportes deben excluir información personal sensible. | Debe contar con Supervisión humana experta en el dominio y en modelamiento | Informes, texto. |

**Desarrollo.**

1. **¿De qué fuentes provienen los datos y cómo se manejan?**

**Los datos provienen de múltiples fuentes clínicas y externas, tales como**

* Historias clínicas electrónicas con información estructurada de pacientes como edad, sexo, antecedentes, diagnósticos previos, resultados de laboratorio, y las notas tomadas por los médicos durante las citas médicas
* Formularios clínicos o autodeclaraciones de síntomas donde los pacientes describen en texto libre cómo se sienten.
* Bases públicas o compartidas de enfermedades raras
* Datos abiertos del sector salud.
* APIs médicas y scraping para recolectar información complementaria sobre síntomas, prevalencias y criterios diagnósticos.

**Manejo de los datos**

* Preprocesamiento estructurado con la codificación de variables categóricas, normalización de valores numéricos.
* Procesamiento de texto (NLP), conversión de síntomas expresados en lenguaje natural a vectores numéricos y datos estructurados
* Imputación de datos faltantes y tratamiento de outliers clínicos.
* Validación de calidad y consistencia antes de alimentar el modelo.
* Anonimización y cumplimiento legal para proteger los datos sensibles del paciente.

1. **¿Qué tipos de modelos de ML se puede usar?**

Dado que el objetivo es clasificar si un paciente puede tener o no una enfermedad (común o huérfana), se recomienda usar modelos supervisados de clasificación, como: Random forest, XGBoost, Regresión Logística o Redes Neuronales Multicapa (MLP) aunque esta ultima muy limitada por la cantidad de datos que se puedan disponer

1. **¿Cómo se van a validar/testear?**

Para validar y testear el modelo predictivo, se implementará una estrategia de validación cruzada estratificada, que permite evaluar su desempeño asegurando una distribución representativa de enfermedades comunes y huérfanas en cada subconjunto. Esta técnica resulta esencial para garantizar que el modelo generalice adecuadamente en escenarios con desbalance de clases. Asimismo, se utilizarán métricas especializadas como el F1-score, la curva ROC-AUC, sensibilidad (recall), con el fin de capturar tanto la capacidad del modelo para detectar casos positivos como para evitar falsos diagnósticos.

**Producción**

1. **¿Cómo se va a desplegar la solución?**

La solución será desplegada mediante un servicio web basado en API, que permitirá integrarla fácilmente con plataformas clínicas o aplicaciones utilizadas por profesionales de la salud. El modelo será empaquetado en un contenedor utilizando herramientas como Docker, lo que asegura su portabilidad y reproducibilidad en distintos entornos, ya sea en servidores hospitalarios o en infraestructura en la nube (según sea el caso). Este despliegue garantizará que el sistema esté disponible para recibir síntomas como entrada y entregar predicciones en tiempo real, facilitando su uso clínico en escenarios de diagnóstico asistido (importante este último).

1. **¿Se necesita algún tipo de monitoreo para esta aplicación?**

Sí, es indispensable implementar un sistema de monitoreo continuo del modelo una vez esté en producción. Este monitoreo está detallado en el pipeline y permitirá detectar posibles caídas en el rendimiento predictivo, cambios en la distribución de los datos de entrada (data drift) o el surgimiento de nuevos patrones clínicos que el modelo no haya visto durante el entrenamiento. Además, se registrarán métricas clave como las descritas previamente además de los tiempos de respuesta, lo anterior servirá para generar alertas si estas métricas se desvían de los umbrales definidos por los equipos del dominio y de datos. El monitoreo también incluirá trazabilidad de las predicciones y el registro de logs para auditar el comportamiento del sistema ante cada solicitud.

1. **¿Pueden existir nuevos datos necesarios quizás para re-entrenar el modelo?**

Sí pueden existir nuevos datos que sirvan para reentrenar el modelo, en un entorno clínico real es esperable que se generen constantemente nuevos datos de pacientes, especialmente en el caso de enfermedades huérfanas que están en continuo estudio. Estos datos, una vez validados y anonimizados, pueden alimentar el pipeline de reentrenamiento del modelo para mejorar su precisión y adaptabilidad.