Pipeline propuesto (MLOps)

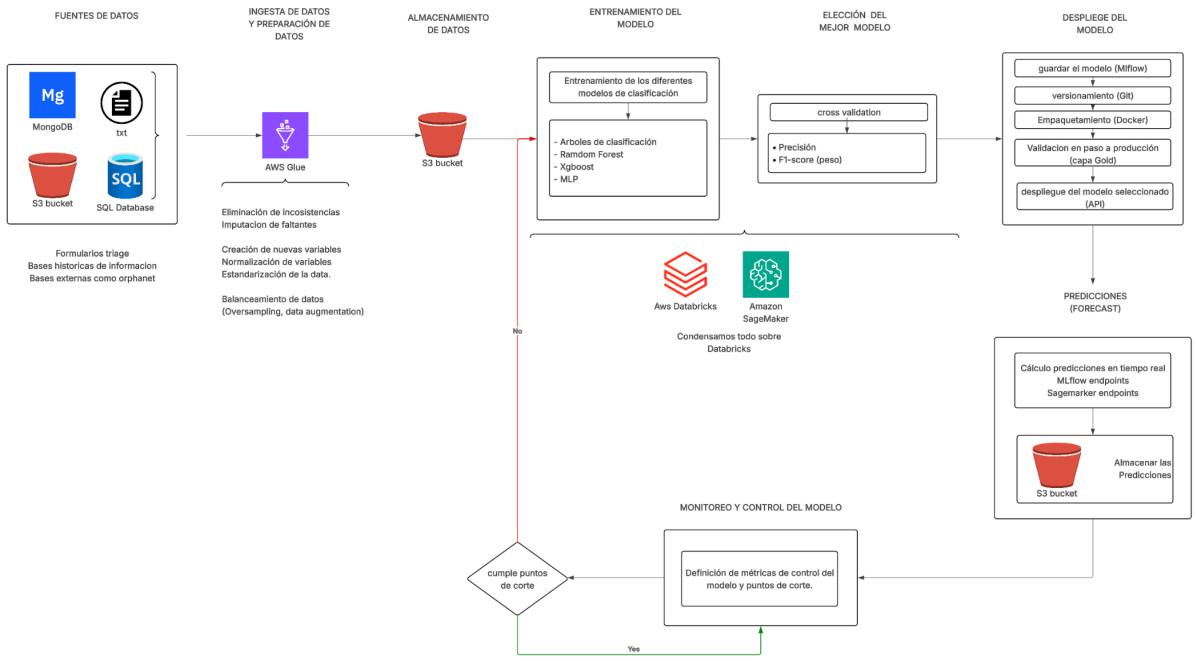
Presentado por:
Jonathan Pacheco

Definición del problema.

Dados los avances tecnológicos, en el campo de la medicina la cantidad de información que existe de los pacientes es muy abundante. Sin embargo, para algunas enfermedades no tan comunes, llamadas huérfanas, los datos que existen escasean. Se requiere construir un modelo que sea capaz de predecir, dados los datos de síntomas de un paciente, si es posible o no que este sufra de alguna enfermedad. Esto se requiere tanto para enfermedades comunes (muchos datos) como para enfermedades huérfanas (pocos datos).

Objetivo:

Diseñar un pipeline de aprendizaje automático que permita a partir de los datos del paciente predecir la clasificación de enfermedad huérfana del mismo (clasificación 4 categorías) y que tenga la oportunidad de actualizar el modelo cada vez que pierda relevancia en el poder de clasificación.



---

1. Fuentes de datos:
    - Formulario triage
    - Historia clínica del paciente
    - Base externa como Orphanet

2. Ingesta de datos y preparación de datos

    - LIMPIEZA DE DATOS
      tratamiento de errores en datos
      tratamiento de faltantes
      recategorizaciones de datos

    - CREACIÓN DE VARIABLES
      Creación de variables categóricas
      Normalización
      Estandarización

    - BALANCEAMIENTO DE CLASES
      Oversampling
      Data augmentation

3. Almacenamiento de datos:

    Almacenar el preprocesamiento de datos en S3

4. Entrenamiento de modelo

    PROPUESTOS
    - Árboles de decisión
    - Random forest
    - XGboost
    - MLP

5. Elección del mejor modelo

- Validación cruzada
- Métricas:
  Precisión
  F1 score

6. Despliegue de modelo
   - Almacenar el modelo en Mlflow
   - Manejo de versiones con git.
   - Contenerización para la reproducción del modelo Docker

   Paso a producción.
   - Validación en ambiente productivo
   - Validación de casos de uso
   - Ajuste de errores
   - Garantizar el ambiente productivo

   Despliegue del modelo.
   - FastAPI, databricks, Sagemaker.

7. Predicciones.
   Cálculo de valores en tiempo real Ml flows endpoint.
   Entrada: Datos clínicos del paciente.
   Salida: Categorías de Clasificación para enfermedad huérfana.
   Almacenamiento de los resultados en S3.

8. Monitoreo y control del modelo.
   Al final de cada mes, cuando se encuentre cargada la clasificación real del cliente en S3 debido a la continuidad del negocio, se evaluará el rendimiento del modelo.

   Monitoreo continuo del modelo.
   - Métricas para el monitoreo del modelo (precisión, recall).

   Definición de puntos de corte de aceptación para métricas de aceptación del performance del modelo.

9. Reentrenamiento y almacenamiento de modelos históricos.

   - Mejora continua del modelo
   - Almacenamiento de versiones de modelos
   - Despliegue de nuevo modelo
   - CI/CD para automatizar el proceso completo de la construcción del modelo.

Nota:
Paralelo al pipeline tener el cuenta los servicios para tener un mayor control.

IAM el cual controla los permisos a los recursos de Amazon, quien tiene acceso y si el acceso es de lectura o escritura.

Cloudtrail: registra quien ejecutó funciones, que usuario accedió y a cual recurso.

Cloudwatch monitorea el comportamiento, latencia del api gateway, memoria en EC2 y otras cosas más en todo el pipeline.