Informe para auditoria: Big Eye BA, Sprint 1

Objetivos Iniciales.

- ✅ Investigación de infraestructura MLOps (con MLFlow y Prefect).
- ✅ Despliegue de infraestructura MLOps (con MLFlow y Prefect).
- ✅ Investigación del Catastro como fuente de datos.
- ⚠️ Creación de un dataset con datos del Catastro.
- ❌ Investigación sobre modelos de detección (usando datos del Catastro).
- ❌ Creación de un modelo básico de detección (usando datos del Catastro).
- ✅ Investigación del proyecto Copérnico y las misiones SENTINEL como fuentes de datos.
- ✅ Creación de un dataset con datos de Copérnico y SENTINEL.
- ⚠️ Investigación sobre modelos de segmentación (usando datos de Copérnico).
- ⚠️ Creación de un modelo básico de segmentación (usando datos de Copérnico).

✅ Completado con éxito.

⚠️ Surgieron problemas.

❌ No se completó.

Problemas que han surgido.

- Dificultades a la hora de obtener datos del Catastro: se llega al problema de que desde la web es muy complicado y poco accesible, además del formato XML que no permite la descarga directa. El servicio de descarga no parece estar disponible los fines de semana.
    - Solución: Elección del software QGis con el addon Inspire Spanish Catastral: La mejor alternativa que se encontró fue el programa QGis para el manejo y visualización de los datos junto con el addon Inspire Spanish Catastral con el que se puede descargar un dataset de prueba, para analizar su contenido y extraer en un dataframe polygon geometries que serán los que usaremos en el modelo.
- Los datos del Catastro tenían una naturaleza distinta a la esperada.
    - Solución: Se planteó realizar un modelo de detección de instancias (edificios), pero dado que ofrece datos sobre la geometría de estos, se opta por crear un modelo de segmentación.
- No ha sido posible crear un modelo usando datos del Catastro. El retraso en la obtención de estos datos impidió que el equipo realizara la investigación sobre el modelo asociado y su creación.
- Los datos de Copérnico tenían una naturaleza distinta a la esperada: Se pretendía hacer un modelo de segmentación que midiera superficies (con vegetación, nieve, agua, urbanas). No obstante, algunas de las bandas ya ofrecían esa información y, además, los datos no tienen polígonos asociados.
    - Solución: Se planteó un modelo de predicción de series temporales usando TCN, de momento, para el índice de vegetación. Podría utilizarse más adelante para tareas de diagnóstico.

Detalle de las tareas finales realizadas.

- Investigación de infraestructura: De acuerdo con las recomendaciones del profesor se estudian las tecnologías de MLFlow y Prefect como herramientas para monitorizar el ciclo de vida de los modelos de machine learning y de orquestación del flujo de trabajo respectivamente.
- Despliegue del servicio de MLFlow: Se configura un servicio básico de MLFlow y se crea un dominio dinámico usando no-ip para que sea accesible.
- Configuración del entorno de Prefect: Después de investigar las opciones de tipos de trabajadores que ofrece Prefect, configuramos work-pools de tipo proceso y usando imagen de Docker. En principio usaremos las de tipo proceso que permiten distribuir el trabajo usando la API de Prefect y son más sencillas.
- Investigación de los datos del catastro: Buscar los tipos de datasets que este ofrece. Después buscar otras alternativas.
- Descargar de un dataset: Finalmente se pudo descargar un dataset de 5 muestras compuestas de varios archivos GML y otros tipos. A partir de estos se obtendrán más muestras, puesto que el archivo de Madrid es muy grande.
- Visualización y extracción de la mejor capa: Cargar todas las capas de cada modelo superpuestas en QGis para ver cual tiene la información más relevante. Extraer esa capa para posteriormente convertirla a GeoJson.
- Investigación de los datos que ofrece el programa Copérnico: Se investigan las distintas misiones Sentinel del programa y su naturaleza, hasta llegar a la conclusión de que el modelo adecuado para nuestras necesidades es el Sentinel-2 con el algoritmo L2A. Se prueban las distintas APIs como el caso de Sentinel Hub sin conseguir el alcance deseado, hasta llegar a la conclusión de que la mejor opción es OpenEO.
- Investigación sobre la API de OpenEO y los datacubes: Se experimenta con la librería en un notebook de Python sobre las distintas funcionalidades ofrecidas, hasta dar con los Datacubes, que son estructuras de datos parecidas a los hipercubos espaciales con dimensión temporal, perfectos para nuestro proyecto. Se decanta finalmente por la búsqueda de cambios en la flora mediante el cálculo del NDVI (Normalized Difference Vegetation Index) usando las bandas B08 y B04 mediante la fórmula $NDVI = \frac{B_8 - B_4}{B_8 + B_4}$
- Creación de una función genérica para la obtención de datos: Se obtienen datos con mayor frecuencia mediante esta función, que permite especificar parámetros espaciales, temporales y las distintas bandas que ofrecen los distintos satélites, que en nuestro caso nos basta con el L2A. Con ella se crea un pequeño dataset de prueba. Se muestran los datos por las bandas RGB.
- Preprocesado de los datos: Se convierten los datos a un array de Numpy a raíz de los Datacubes de OpenEO. A continuación, se divide el hipercubo en cada cubo correspondiente a un instante temporal. De cada cubo se obtiene la capa de NDVI y se calcula su media. El resultado es una serie temporal con las medias del índice NVDI para cada instante.
- Creación de un modelo de predicción de series temporales: Se utiliza la arquitectura TCN para crear un primer modelo de prueba capaz de predecir la evolución del índice NVDI medio. Se realiza ya utilizando la infraestructura MLOps.