Módulo 1: Moving from Research to Production

Adopting an MLOps mindset

MLOps

Es el proceso de automatizar el flujo trabajo y el ciclo de vida de los proyectos de Machine Learning, desde la experimentación hasta la puesta en producción.

También, asegura que los experimentos de ML fueron correctamente probados y están listos para desplegarse y escalar.

ML experiments

Los experimentos en ML involucran probar diferentes modelos y determinar cuál es el mejor.

MLOps incluye: experimentación con los modelos; evaluar diferentes modelos en diferentes conjuntos de datos; llevar una selección cuidadosa. El proceso de selección puede consumir mucho tiempo, pero es un paso crucial en cualquier proyecto.

Un experimento de ML está listo para llevarse de la fase de experimentación a producción cuando:

- Se ha documentado apropiadamente.
- Se ha probado y validado arduamente para asegurar la precisión y la confiabilidad.
- Se ha asegurado que el ambiente es seguro y escalable.

Why most ML experiments fail

- Falta de metas y objetivos claros.
- Pobre calidad en los datos.
- Arquitecturas de modelos complejas.
- Datos de entrenamiento insuficientes.
- Overfitting o underfitting.

Technical debt

La deuda técnica ocurre cuando se escribe código de forma apresurada, sin realizar pruebas y validaciones, produciendo errores que pueden ser costosos y requerir de bastante tiempo para resolverlos. También se puede producir por documentación desactualizada o faltante en cualquier proceso de codificación o selección de modelos de ML.

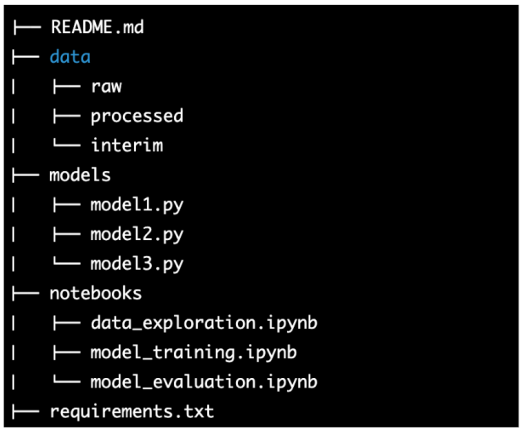
Para evitar incurrir en la deuda técnica es esencial priorizar la calidad y correctitud del código y la documentación. Siguiendo buenas prácticas y tomando el tiempo necesario para probar y validar el código y los modelos de ML de forma adecuada, podemos asegurar que nuestros proyectos sean exitosos y sostenibles en el largo plazo.

Writing maintainable ML code

Project structuring

El primer paso en crear código en ML que sea mantenible es organizar los archivos del proyecto en una estructura lógica; esto significa agrupar los archivos relacionados en carpetas separadas, como los conjuntos de datos o los modelos de ML. También, es importante asegurarnos que los archivos están debidamente nombrados y etiquetados para que sea fácil localizarlos.

Sample project directory with a README file, requirements file and three subfolders: data, models, notebooks



- README.md: Explains the purpose of the repository and how to use it.
- requirements.txt: Lists all dependencies
- data: contains data-related files, including raw data and processed data
- models: contains all model-related files, including scripts for creating models.
- notebooks: contains notebooks for data exploration, model training, and model evaluation.

Code versioning

Una manera de hacer que el código sea mantenible es utilizar un sistema de control de versiones como Git. Esto nos permite llevar un seguimiento de los cambios realizados al código; deshacer cambios y devolvernos a versiones anteriores rápidamente; identificar la fuente de errores y problemas para corregirlos fácilmente; permitir que varios desarrolladores trabajen en paralelo.

Documentation

La documentación del código y la estructura del proyecto es otra parte esencial. Incluye la explicación de cada archivo del proyecto, cómo usar el código y cómo desplegar el modelo de ML.

Adaptability of code

Uno de los beneficios clave de un código mantenible es que es fácil de entender, modificar y actualizar. También, reduce el tiempo y esfuerzo necesario para realizar cambios en el código fuente y minimiza la aparición de errores. Además, es necesario para la construcción de aplicaciones de Ml que puedan evolucionar y adaptarse a través del tiempo.

Writinf effective ML documentation

The components of excellent ML documentation

Hay 6 áreas principales en la documentación:

- Data sources
- Data schemas
- Labeling methods
- Model experimentation + selection
- Training environments
- Model pseudocode

Data sources

Documentar las fuentes de los datos nos permite establecer procesos para evaluar la calidad de nuestros datos, proporcionando una base de comparación para identificar errores potenciales o inconsistencias. También, nos ayuda a:

- Llevar un seguimiento de dónde vienen los datos
- Evaluar e iterar la calidad de los datos

Data schemas

El esquema de los datos es la estructura que describe la organización de los datos .

Database key	Data type	Data order
`Person.name`	`string`	`nominal`
`Person.survey_score`	`integer`	`ordinal`

Por ejemplo, en una base de datos relacional se especificarían las tablas, campos en cada una y las relaciones entre campos y tablas.

Escribiendo esquemas en nuestra documentación podemos proporcionar una estructura de los datos que de otra manera sería desorganizada, además de hacerle saber a los demás qué clase de datos está utilizando el modelo para aprender.

Labeling methods (for classification)

Cuando trabajamos con problemas de clasificación queremos documentar cómo llegamos a las etiquetas finales de la variable respuesta.

Por ejemplo, cuando trabajamos con datos crudos y no estructurados como imágenes, es posible que no hayan sido previamente etiquetados.

Entendiendo exactamente cómo los datos fueron recolectados y etiquetados es vital para la reproducibilidad. También podemos utilizar esto para asegurar la calidad de los datos, lo que en última instancia afectan a la confiabilidad y desempeño del modelo.

Model pseudocode

Es una representación simplificada de los pasos involucrados en la construcción del modelo de ML.

A menudo incluye escribir:

- Los pasos del proceso de feature engineering
- Los componentes del pipeline de orquestación
- Entradas y salidas esperadas por el modelo

Model experimentation + selection

Documentar el proceso de experimentación y selección del mejor modelo de ML. Se incluye:

- El proceso de desarrollo del modelo
- Los modelos considerados
- Las métricas usadas en la evaluación y selección
- La combinación de hiperparámetros usadas para cada modelo

Training environments

Además de documentar el proceso de experimentación y selección también debemos hacerlo con el ambiente que utilizamos, incluyendo cualquier paquete de terceros o librerías, semillas para valores aleatorios.

Este paso es crucial para asegurar la reproducibilidad y consistencia de los resultados del entrenamiento del modelo de ML y el despliegue a producción.