Laboratorio 1
Gestión de Experimentación y Modelos con DVC
Automated Machine Learning (AutoML)

Objetivo:

El objetivo es diseñar un pipeline automatizado usando DVC (Data Version Control) para gestionar datasets, modelos y experimentos con diferentes configuraciones. Este flujo debe reflejar los principios de AutoML, siendo capaz de adaptarse a cualquier dataset estructurado, entrenar múltiples modelos, y comparar su rendimiento mediante preprocesamiento automático y configuración dinámica de hiperparámetros. La idea es fomentar la reproducibilidad, eficiencia, y gestión efectiva de modelos y datos en proyectos de ciencia de datos.

Descripción del Problema:

El laboratorio se centrará en construir un pipeline flexible que aplique los principios de AutoML. A través de este pipeline, se podrán aplicar transformaciones automáticas según los tipos de variables del dataset (e.g., normalización para variables numéricas y codificación OneHot para categóricas), entrenar múltiples modelos y gestionar experimentos de manera eficiente. El flujo debe estar diseñado para funcionar con cualquier dataset estructurado (como finalidad únicamente de pruebas pueden utilizar el dataset de housing que se encuentra en el GES). La idea central del laboratorio es permitir experimentar, optimizar, y comparar modelos automáticamente.

El enfoque de AutoML es relevante en la práctica porque:

- **Ahorra tiempo**: Automatiza tareas repetitivas y reduce el esfuerzo manual en la experimentación.
- **Mejora la productividad**: Facilita la comparación de modelos y el ajuste de hiperparámetros de manera sistemática.
- **Reduce la curva de aprendizaje**: Permite que tanto expertos como principiantes trabajen de manera eficiente.
- **Facilita la toma de decisiones informada**: Provee un enfoque reproducible para experimentar con diferentes estrategias y seleccionar la más adecuada.

Este laboratorio busca emular estos beneficios mediante un pipeline basado en DVC que permita la gestión de modelos, datos y experimentos de manera sencilla y organizada.

Requerimientos:

1. **Exploración y Visualización de Datos**
    - Cargar los datos del archivo correspondiente.
    - Verificar si hay valores faltantes o errores en los datos, y aplicar las correcciones necesarias.

2. **Preprocesamiento de Datos**
    - Aplicar transformaciones adaptativas según los tipos de variables definidos en `params.yaml` (e.g., normalización para numéricas, OneHot para categóricas, etc.).
    - Dividir el dataset en conjuntos de entrenamiento y prueba según los parámetros definidos.
3. **Selección y Entrenamiento de Múltiples Modelos**
    - Implementar varios modelos de clasificación o regresión basados en el tipo de problema del dataset seleccionado.
    - Entrenar automáticamente múltiples algoritmos de machine learning (e.g., Regresión Lineal, Random Forest, Gradient Boosting).
    - Utilizar los modelos y sus configuraciones definidas en `params.yaml`.
    - Almacenar los modelos entrenados en un repositorio versionado para facilitar su gestión.
    - Mostrar los resultados de los modelos.
    - El autoML debe ser capaz de indicar cual es el mejor modelo para el problema.
4. **Optimización y Validación Cruzada**
    - Implementar una búsqueda automática de hiperparámetros (grid search o random search) para cada modelo configurado.
    - Aplicar técnicas de validación cruzada para evaluar los modelos y mejorar su rendimiento.
    - Comparar los resultados obtenidos y seleccionar automáticamente la mejor combinación de hiperparámetros.
5. **Interpretación de Resultados**
    - Calcular automáticamente las métricas de rendimiento relevantes (e.g., precisión, MSE, F1-Score).
    - El pipeline debe identificar las características más importantes según los modelos utilizados.
    - También debe indicar los parámetros finales de entrenamiento del modelo seleccionado.
    - Exportar los resultados a formato CSV o Markdown para su análisis.

Puntos Extra (Opcional):

1. **Optimización de Hiperparámetros con Optuna**
    - Integrar Optuna como optimizador para la búsqueda de hiperparámetros en uno o varios modelos.
    - Definir un estudio de Optuna que explore diferentes combinaciones de hiperparámetros de manera eficiente.
    - Almacenar los resultados de Optuna y seleccionar la configuración con las mejores métricas.

- Mostrar las curvas de optimización para visualizar la convergencia hacia los mejores hiperparámetros.

2. **Optimización del Tiempo de Entrenamiento**
   - Implementar técnicas para reducir el tiempo de entrenamiento de los modelos, como **early stopping** o **entrenamiento en paralelo**.
   - Evaluar el impacto del tiempo de entrenamiento en la calidad del modelo y discutir las compensaciones entre tiempo y precisión.

Entregables:

- Un archivo .zip que contenga:
  - Instrucciones claras sobre cómo reproducir el proyecto, incluyendo los comandos necesarios para ejecutar el pipeline de DVC.
  - El código completo del proyecto, incluyendo todos los archivos necesarios para ejecutar el pipeline con DVC y demostrar el trabajo realizado en cada uno de los requerimientos anteriores.
  - El archivo .zip debe ser descargado desde el **release o tag** generado en GitHub, garantizando que la versión entregada esté congelada en el momento del release.
  - En el GES, además de subir el archivo .zip, deben incluir el **link al release o tag en GitHub**.