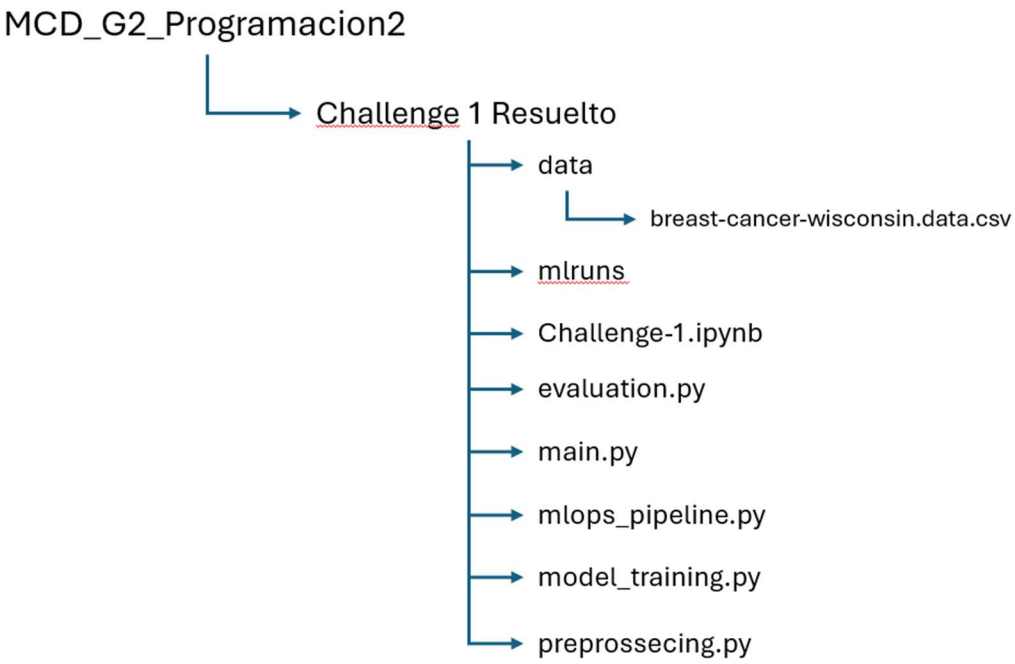
Reto 1 - Clasificación de tumores de mama

El cáncer de mama es una de las principales causas de mortalidad en mujeres a nivel mundial, lo que hace que la detección temprana y el diagnóstico preciso sean fundamentales para salvar vidas. En este contexto, la ciencia de datos desempeña un papel crucial, ya que mediante el análisis de grandes volúmenes de datos, es posible crear modelos predictivos que ayuden a los profesionales médicos a identificar y clasificar los tumores de mama de manera más eficiente.

A partir de un conjunto de datos que contiene información sobre características numéricas extraídas de imágenes de células tumorales, incluyendo medidas como el radio, la textura, el perímetro, el área, entre otros. El objetivo es clasificar los tumores en dos categorías: malignos (cancerígenos) y benignos (no cancerígenos).

Estructura del Proyecto

El proyecto está almacenado en github para poder acceder a él. En el paquete de información se encuentra la siguiente estructura:



El dataset se encuentra dentro de la carpeta *data*. Mientras que la carpeta de *mlruns* almacenará los parámetros de cada experimento que se haya corrido.

El archivo principal es *main.py*. Este mandará llamar las librerías necesarias para correr el programa sin problemas.

Las librerías de soporte son:

- *Evaluation.py* que es el módulo que evalúa el desempeño del modelo de clasificación.
- *Model_training.py* es el módulo que utiliza el subconjunto de datos de entrenamiento para enseñarle al modelo.
- *Preprocessing.py* se encarga de limpiar el conjunto de datos y estandarizar los valores de las variables.
- *Mlops_pipeline* nos ayuda a llevar un registro de todos las configuraciones y resultados utilizados en cada experimento.

Instrucciones para clonar el repositorio

En la terminal utilice los siguientes comandos:

1. git clone https://github.com/JZGlez/MCD_G2_Programacion2.git
2. cd MCD_G2_Programacion2
3. git sparse-checkout init --cone
4. git sparse-checkout set /Challenge 1 Resuselto

Estos pasos te permitirán generar un clon local del folder donde podrán hacer los cambios y pruebas que desee sin afectar los archivos originales.

Pasos para ejecutar el código

Para poder correr este programa, asegúrate de que ya lo hayas descargado completamente como se describió en la sección anterior. Una vez hecho esto sigue los siguientes pasos:

1. Abre una nueva terminal
2. Accede a la carpeta donde guardaste el proyecto, hasta llegar a la carpeta *Challenge 1 Resuelto*
3. Una vez ahí, corre el archivo *main.py* utilizando la instrucción *Python*

Documentación

En la carpeta *data* se encuentra el archivo *breast-cancer-wisconsin.csv* el cual contiene el conjunto de datos conformado por la información obtenida a partir de las imágenes tumorales. Este conjunto de datos presenta 569 registros, donde cada uno de ellos tiene información sobre las imágenes tales como:

1. ID: Un identificador único para cada observación en el conjunto de datos.
2. Diagnosis: El diagnóstico del tumor, que puede ser: M (Maligno) o B (Benigno)
3. Radius: El radio (o radio medio) de las células tumorales. Se refiere a la distancia desde el centro de la célula hasta su borde.

4. Texture: La textura de las células tumorales, medida a través de la variabilidad de la intensidad de los píxeles en la imagen de la célula.
5. Perimeter: El perímetro de la célula tumoral.
6. Area: El área total ocupada por la célula tumoral.
7. Smoothness: La suavidad de los bordes del tumor. Se mide a través de la variabilidad local en el radio.
8. Compactness: Un valor que combina el perímetro y el área, que describe la compactación de la célula.
9. Concavity: La concavidad de los contornos de la célula, es decir, la profundidad de las depresiones en los bordes del tumor.
10. Concave points: El número de puntos cóncavos en el contorno del tumor, que indica irregularidades en los bordes.
11. Symmetry: La simetría del tumor. Mide la uniformidad en la distribución de la forma y el tamaño del tumor.
12. Fractal dimension: Una medida que describe la complejidad de la forma del tumor, basada en la relación entre su tamaño y su área.

Los valores de estas características están calculados de manera media (mean), desviación estándar (standard error) y mediana (worst). El conjunto de datos contiene una combinación de estas métricas

- **Mean**: La media de las características mencionadas (X_mean).
- **Standard Error**: La desviación estándar de esas características (X_se).
- **Worst**: El valor más alto observado para cada característica (X_worst).

Por ejemplo, para el radio, encontrarás las columnas:

- *radius_mean*: El valor medio del radio.
- *radius_se*: La desviación estándar del radio.
- *radius_worst*: El valor más alto observado del radio.

Y lo mismo ocurre para las demás características (perímetro, área, textura, etc.)

Con estos datos, y tras su preprocesamiento se utilizaron varios métodos, pero el que mejor resultado dio es el *random forest*. Los diferentes modelos se pueden observar dentro del módulo *model_training.py*.

Un **algoritmo de Random Forest** es un modelo de aprendizaje supervisado utilizado para clasificación y regresión. Se basa en un conjunto de **árboles de decisión** entrenados con diferentes subconjuntos de los datos y características. Los árboles se combinan para hacer predicciones, y el resultado final es determinado por el **voto mayoritario** (para clasificación) o el **promedio** (para regresión).