

Estructura general

- fairtest.py recibe el Dataloader con todos los datos. => `load_data(dataloader)`.
- Pasando un label como atributo sensible, permite hacer las pruebas correspondientes => `test(lista_labels)`.
 - Almacena las predicciones que devuelve cada prueba.
- Ofrece dos funciones `standard_metrics()` y `fair_metrics()` que devuelven una matriz con todas las métricas para cada método, además de dos listas con el orden/nombre de métodos y de métricas.
- En `examples\`, va a haber diferentes ejemplos obteniendo el dataloader a partir de diferentes fuentes, y llamando a `fairtest`.
- `fairtest` llama a los diferentes métodos de la carpeta `\methods`.
 - Todos los métodos heredan `method`, que trae `load_data(dataloader)`, `fit(protected_label, protected = true)`, `predict()`.
 - La variable `protected` en la llamada a `fit()` sirve para configurar si la variable protegida debe verse o no en el entrenamiento. Esto en meta-learning no aplica, obviamente.
- En `\metrics`, se implementan las clases `StandardMetrics` y `FairMetrics`, que, dada la tabla de resultados y la variable protegida (en `FairMetrics`), ofrecen:
 - Métodos individuales para cada métrica.
 - Un método que devuelve todas las métricas con el vector de nombres de cada métrica.