

## Atelier 3 : Création d'un Makefile

**Prérequis :** Code modulaire fonctionnel.

1. **Objectifs :** Simplifier l'exécution des tâches avec un Makefile.
2. **Contenu du Makefile :**
  - I. **Installation:**  
« pip install -r requirements.txt »
  - II. **Exécution des étapes liées au modèle :**  
Exemple « python main.py --train »
  - III. **Étapes CI (qualité du code, format du code, sécurité du code, etc ...)**
3. **Livrables :** Un Makefile permettant d'automatiser les principales étapes.

## Tutoriel : Création d'un Makefile pour Automatiser les Tâches du projet

### Introduction :

Un **Makefile** simplifie et automatise les tâches répétitives dans un projet, comme la préparation des données, l'entraînement d'un modèle ou son évaluation. Il joue un rôle similaire à un **Jenkinsfile**, mais est utilisé localement pour orchestrer des workflows sans infrastructure CI/CD. Cela garantit un pipeline clair, reproductible et facile à exécuter via des commandes simples.

Dans notre projet, makefile va intervenir pour à la fois ordonner les étapes liées au modèle (prepare, train, validate, etc....) et les étapes liées au projet en globalité (code, deployment, suivi, etc...)

### I. Étape 01 : Préparer l'environnement

#### 1) Créer un fichier Makefile

- a. Ouvrir le terminal WSL et accéder au répertoire de votre projet.
- b. Créer le fichier « Makefile » :

Le fichier Makefile sera vide. Suivre les étapes ci-dessous pour y ajouter des commandes.

### II. Étape 02 : Définir les sections du Makefile

#### 1) Installer les dépendances

Créer un environnement virtuel et installer les dépendances à partir de requirements.txt.

#### 2) Vérification du code

Ajouter des étapes ci pour la vérification de votre code python (formatage automatique, qualité du code, sécurité du code, etc  
....)

#### 3) Préparer les données

Exécuter la fonction prepare\_data() dans main.py.

#### **4) Entraîner le modèle**

Exécuter la fonction `train_model()` dans `main.py`.

#### **5) Exécuter les tests**

Lancer quelques tests basiques.

**NB :** le jour de la validation, n'importe quelle modification au niveau du dossier de votre projet (fichiers python, makefile, fichier data, etc..) devra lancer automatiquement toutes les étapes présentes dans votre makefile (ci-cd). Pensez à réaliser cette tâche à l'avance avant la séance 5 du cours (tache notée dans la grille finale)

#### **Piste d'excellence :**

Vous pouvez utiliser l'un des outils d'orchestration mentionnés dans le cours au lieu du makefile pour l'automatisation du pipeline.

Vous pouvez créer des pipelines distinctes pour les différentes parties (data, training, code, etc.....)