

# TP Git/GitHub

## Collaboration en binôme avec GitHub

**Livrable :** Compte rendu avec prises d'écran et explications.

### 1. Mise en place du projet

#### a. Etudiant A :

- Créer un nouveau dépôt GitHub nommé TP\_Github
- Cloner le dépôt :  
i. `git clone <URL>`
- Copier les fichiers du projet (main.py, preprocess\_data.py, train\_model.py).
- Pousser le dépôt sur Github :  
i. `git add *`  
ii. `git commit -m "première collaboration Github"`  
iii. `git push origin`

#### b. Etudiant B :

- Cloner le dépôt :  
i. `git clone <URL>`

### 2. Création de branches et modifications

#### a. Etudiant A :

- Créer une nouvelle branche et modifier le fichier `train_model.py`, ensuite effectuer un commit.
- Créer un nouveau fichier texte expliquant vos modifications et sauvegarder en local avec un commit.
- Fusionner les commits avec `git rebase -i origin/main`
- Pousser vos modifications sur le dépôt distant :  
i. `git push origin nom_de_votre_branche`

#### b. Etudiant B :

- Créer une nouvelle branche et modifier le fichier `preprocess_data.py`, ensuite effectuer un commit.
- Créer un nouveau fichier texte expliquant vos modifications et sauvegarder en local avec un commit.
- Fusionner les commits avec `git rebase -i origin/main`
- Pousser vos modifications sur le dépôt distant :  
i. `git push origin nom_de_votre_branche`

### 3. Gestion de conflits

- Modifier la même ligne du fichier `main.py` et sauvegarder en local vos modifications avec des commits.
- L'étudiant A met à jour sa branche locale : `git rebase origin/main` → Mise à jour réussie.
- L'étudiant A pousse ses modifications vers le dépôt distant et les fusionne avec une Pull Request.
- L'étudiant B met à jour sa branche locale : `git rebase origin/main` → Conflit.
- Résoudre le conflit et finaliser la fusion des modifications.