# TP mlflow

**Prise en main et tracking**

1.Installez le package mlflow avec python dans un environnement virtuel.

2.Dans un terminal lancez un serveur mlflow. Aller voir dans votre navigateur, sur le port correspondant, l'ui de mlflow.

3.Dans un notebook, utilisez le package mlflow pour vous connecter au serveur mlflow que vous avez lancé. Utilisez la bonne fonction pour paramétrer l'adresse du serveur.

Attention ! A chaque fois que vous effectuerez une opération sur mlflow dans une autre cellule de votre notebook vous devrez vérifier avant que vous pointez bien sur le bon serveur mlflow. Il existe aussi une fonction pour connaître quelle adresse de serveur a été enregistrée.

Créez une experiment via votre notebook ou avec via l'ui.

Une experiment, ou une expérience en français, est un ensemble de run que vous avez effectué. Le but est de trouver in fine les meilleurs paramètres, modèles ou hyperparamètres pour un besoin.

4.Après avoir créé l'experiment. Nous allons entraîner notre modèle. Prenez le dataset wine de la librairie sklearn et utilisez un algorithme de la famille des arbres de décisions. Nous allons lors de l'entraînement de notre modèle logger les mesures.

Pour cela nous voulons lancer un nouveau run dans notre experiment sur mlflow. Le run correspond à un entraînement du modèle.

-Nommez le nouveau run que vous voulez créer avec la date du jour, l'heure, la minute et la seconde.

-Trouvez comment logger les paramètres d'entraînement, les hyperparamètres et les performances du modèle dans mlflow en lançant votre run.

-Ajoutez également votre modèle avec un nom distinctif, vous devrez réutiliser ce nom lors des prochains entraînements. Que voyez-vous dans l'ui de mlflow ?

5.Changez un ou plusieurs hyperparamètres et relancez un entraînement avec le même nom de modèle. Rendez-vous dans l'onglet model de mlflow que constatez-vous ?

6.Entraîner 3,4 modèles avec des hyperparamètres différents.

**DVC**

Pour chaque run le dataset utilisé est précisé dans mlflow. Mais les informations fournies sont assez pauvres. Pour améliorer ça nous allons utiliser l'outil dvc en combinaison avec mlflow.

Suivez les instructions du notebook.

**Déploiement**

Trouver comment transitionner un modèle en état staging et ensuite dans l'état production.

Créer une fonction qui récupère le nom du modèle avec les meilleurs metrics d'entraînement et qui transitionne ce modèle dans l'état production.

Le serveur mlflow peut vous fournir des prédictions à partir des modèles enregistrés. Faites en sorte d'obtenir une prédiction de votre dernier modèle en requêtant le serveur mlflow.

Créer un script à part qui pull le dernier modèle depuis le model registry. Plus tard vous pourrez utiliser ce script pour récupérer le modèle dans une api.

**Serveur centrale**

Demander l'adresse du serveur mlflow au formateur ainsi que vos id de connexion

Entraînez cette fois ci un CNN avec pytorch et enregistrez les paramètres et mesures de performances du modèle.

**(Bonus) Création d'une pipeline d'entraînement avec mlflow et Airflow**

Nous allons utiliser Airflow pour entraîner un modèle de machine learning et mlflow pour enregistrer ses paramètres et performances.

Pour cela récupérer le fichier docker-compose.yml fournit dans la documentation Airflow. Vous devez avoir Docker d’installer sur votre ordinateur.

Notre pipeline récupérera les x lignes d'un dataset et lancera un entraînement à partir de ces données.

Vous pourrez prendre le dataset fourni dans la partie dvc et modifier à chaque fois la période de temps prise.

Ensuite nous testerons les performances du modèle, si le modèle atteint des performances supérieures à un seuil défini alors nous enregistrerons ce modèle dans mlflow (dans le modèle repository).

Vous réaliserez une pipeline de déploiement, qui prend le modèle avec les meilleures caractéristiques et le passe dans l'état production.