2 Mise en place d'une méthodologie MLOps pour le déploiement de modèles de machine learning

2.1 Contexte de travail et Problématique

Dans le cadre de ce projet, ma mission consistait à mettre en place une méthodologie MLOps pour faciliter le déploiement des modèles de machine learning dans la ville de Paris. L'objectif était de créer un outil efficace et automatisé permettant aux équipes de data science de déployer leurs modèles en production tout en assurant leur fiabilité et leur évolutivité.

Pour atteindre cet objectif, j'ai choisi d'utiliser des outils tels que MLflow pour la gestion du cycle de vie des modèles. Cependant, avant de procéder au déploiement des modèles, il était nécessaire de mettre en place un environnement adéquat avec les outils indispensables pour le développement, la gestion des versions et le déploiement des modèles.

Cependant, un des défis rencontrés était le manque de ressources matérielles suffisantes, notamment l'absence de GPU pour les tâches d'entraînement de modèles complexes. Par conséquent, l'équipe a décidé d'installer les outils et de suivre la méthodologie MLOps dès qu'ils disposeront des ressources nécessaires.

2.2 Comprendre MLOps : Fusion de Machine Learning et Opérations

Avant de plonger dans les détails de l'application de la méthodologie MLOps dans ce projet, il est important de clarifier ce que l'on entend par MLOps. Le terme "MLOps" est une contraction de "Machine Learning" et "Operations". Il désigne une approche méthodologique qui vise à combler le fossé entre le développement de modèles de machine learning et leur déploiement en production, en intégrant les meilleures pratiques des opérations informatiques.

L'objectif fondamental du MLOps est de créer un processus continu et collaboratif pour le développement, le test, le déploiement et la gestion des modèles de machine learning. Contrairement aux approches traditionnelles qui se concentrent souvent uniquement sur la phase de développement du modèle, le MLOps met l'accent sur la transition en douceur du modèle de la phase de recherche à la production, tout en assurant la traçabilité, la reproductibilité et la scalabilité.

Au cœur du MLOps se trouvent les concepts de gestion des versions, d'automatisation, de suivi des expérimentations, de déploiement continu et de surveillance des performances des modèles

en production. Cette approche vise à éliminer les silos entre les équipes de développement et les équipes opérationnelles, permettant ainsi une collaboration transparente tout au long du cycle de vie du modèle.

Le MLOps implique également la mise en place d'outils et de processus standardisés qui garantissent que les modèles peuvent être déployés rapidement, de manière reproductible et sans faille. Cela inclut l'utilisation d'outils tels que MLflow, Airflow, FastAPI et d'autres technologies connexes qui facilitent la gestion, la surveillance et le déploiement des modèles.

Dans ce contexte, l'application de la méthodologie MLOps dans ce projet revêt une importance cruciale pour assurer la réussite du déploiement des modèles de machine learning dans un environnement urbain complexe. Passons maintenant à l'examen détaillé de la manière dont les outils tels que MLflow, Airflow, FastAPI et Minio ont été utilisés pour concrétiser cette méthodologie dans l'application pratique.

2.3 Application

Dans cette partie, je décrirai en détail les activités pratiques que j'ai réalisées pour préparer le projet sur GitHub et pour présenter la méthodologie MLOps à l'équipe.

2.3.1 Préparation du projet sur GitHub

J'ai créé un projet sur GitHub dédié à la méthodologie MLOps pour le déploiement de modèles de machine learning. Ce référentiel central a été conçu pour rassembler toutes les informations, la documentation et les ressources nécessaires pour l'installation et l'utilisation des outils MLOps.

2.3.2 Les outils utilisés

Dans la mise en place de la méthodologie MLOps pour le déploiement de modèles de machine learning, j'ai fait usage de plusieurs outils clés qui ont joué des rôles essentiels dans la création d'un processus fluide et automatisé. Chacun de ces outils a été choisi en fonction de sa pertinence pour les besoins spécifiques du projet et de sa capacité à s'intégrer harmonieusement dans l'environnement existant.

2.3.2.1 Mlflow

MLflow a été choisi comme plateforme centrale pour la gestion du cycle de vie des modèles. En plus de son rôle traditionnel de suivi des expériences et de gestion des métriques, nous avons configuré MLflow pour stocker les artefacts générés lors de l'entraînement des modèles. Cette

configuration permet de stocker les modèles entraînés, les métriques de performance, les hyperparamètres et autres informations pertinentes dans un emplacement centralisé. [2]

2.3.2.2 Bentoml

BentoML a été utilisé pour créer des conteneurs de déploiement des modèles, ce qui a simplifié la tâche de transformation des modèles en API de production. En encapsulant les modèles dans des conteneurs, BentoML garantit que les modèles déployés sont portables et indépendants de l'environnement, ce qui facilite leur intégration dans le flux de production. Minio

Minio, un service de stockage d'objets compatible avec S3, a été utilisé pour stocker en toute sécurité les artefacts et les modèles générés par le processus. La configuration de MLflow pour pointer sur Minio a permis de centraliser le stockage des données, garantissant leur disponibilité et leur persistance. [3]

2.3.2.3 Fast Api

FastAPI a été intégré dans le processus de déploiement pour fournir une interface d'API conviviale permettant de servir les modèles déployés. Cette intégration a permis aux utilisateurs finaux d'interagir avec les modèles de manière intuitive et d'obtenir rapidement des prédictions en temps réel. [4]

2.3.3 Architecture

L'architecture complète de la méthodologie MLOps reflète une interaction harmonieuse entre ces outils, ce qui crée un flux de travail rationalisé et automatisé. Voici les étapes clés :

Expérimentation et Entraînement : Dans cette première phase, les scientifiques des données ont utilisé MLflow pour suivre leurs expérimentations et entraîner différents modèles. Les résultats, y compris les métriques et les artefacts, ont été enregistrés dans Minio.

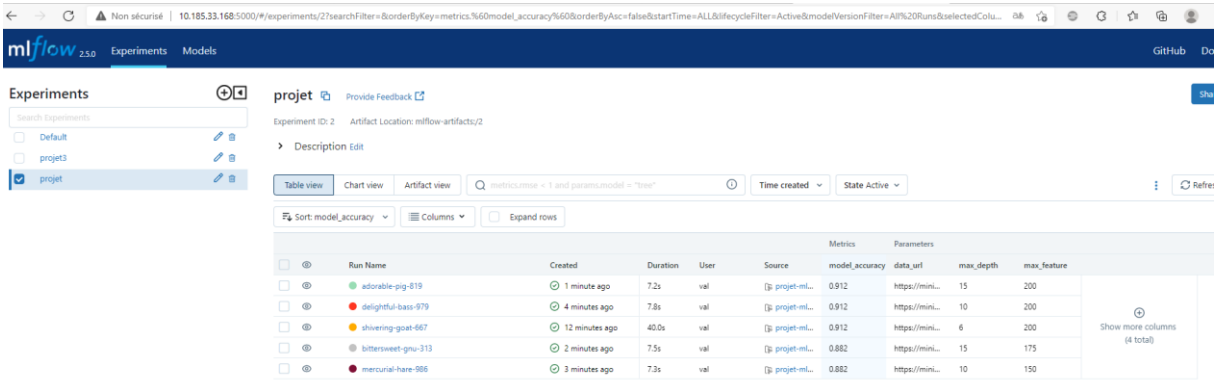

Figure 1: uploader plusieurs modèles de machine learning sur MLFLOW

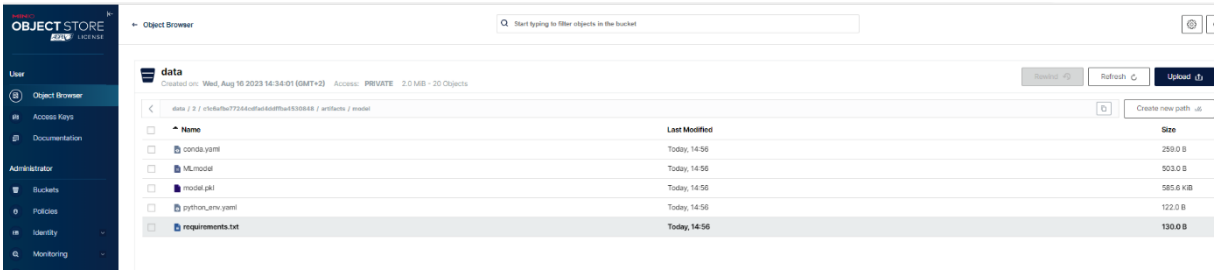

Figure 2:les modèles affichés sur mlflow sont stockés sur Minio

Comparaison et Sélection des Meilleurs Modèles : Après avoir entraîné plusieurs modèles, une phase de comparaison a eu lieu. MLflow a permis de comparer les performances et de sélectionner les meilleurs modèles en fonction des critères prédéfinis.

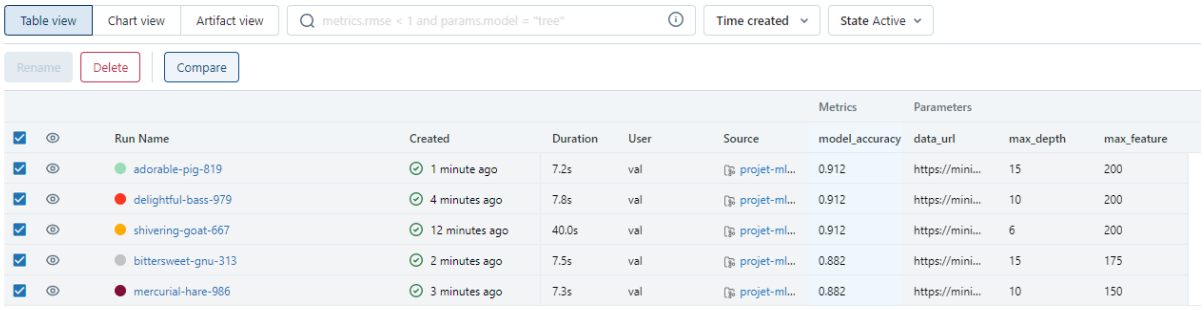

Figure 3:Sélectionner et trier les modèles

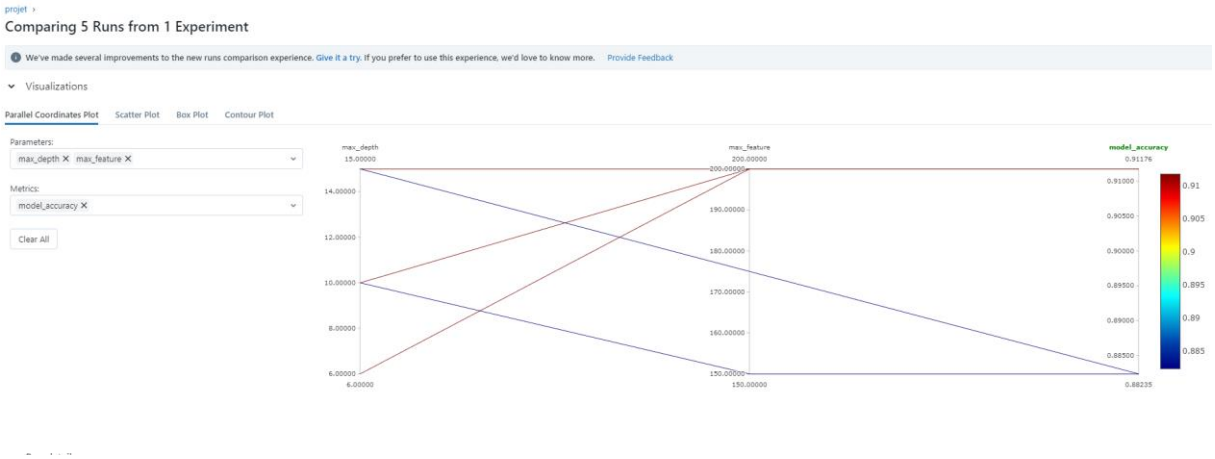

Figure 4: Comparaison des modèles

Enregistrement des Meilleurs Modèles : Les meilleurs modèles ont été enregistrés dans MLflow avec des balises spécifiques pour les identifier plus facilement lors du déploiement.

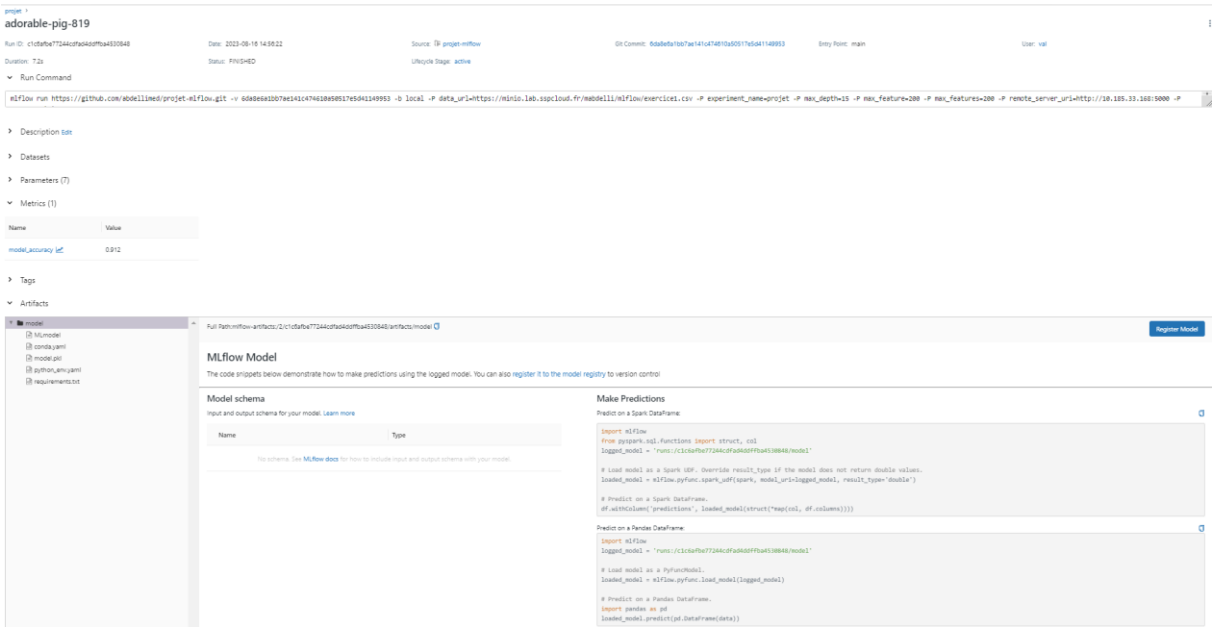

Figure 5: Affichage des métadonnées du modèle

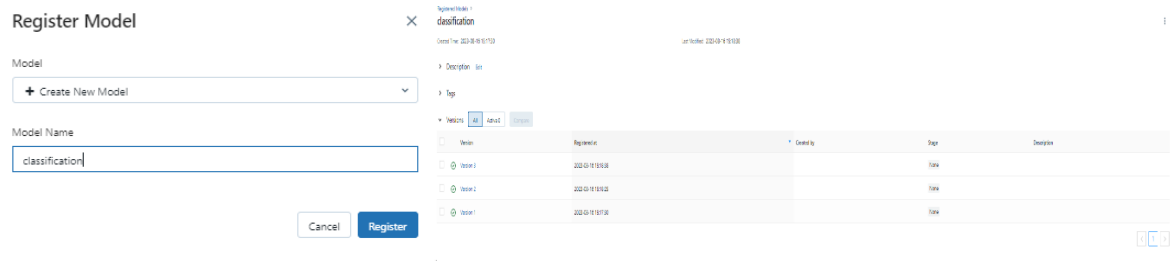

Figure 6: Enregistrer les meilleurs modèles

Transition en Production : La mise en production des modèles sélectionnés a été initiée. MLflow a joué un rôle clé en détectant les modèles enregistrés et en les mettant à disposition pour le déploiement.

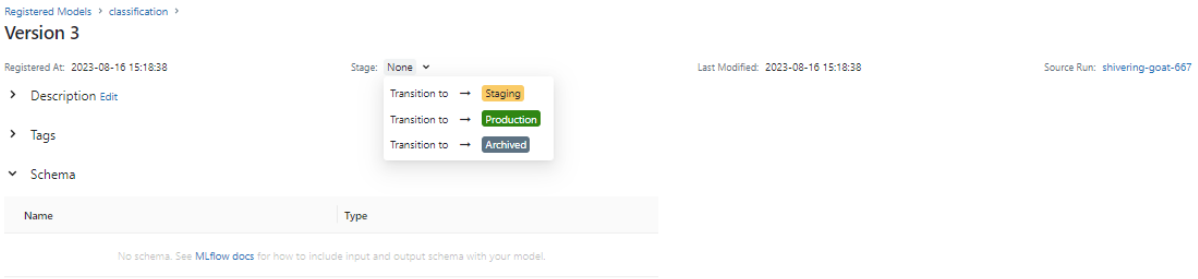

Figure 7 : Mettre un modèle en production

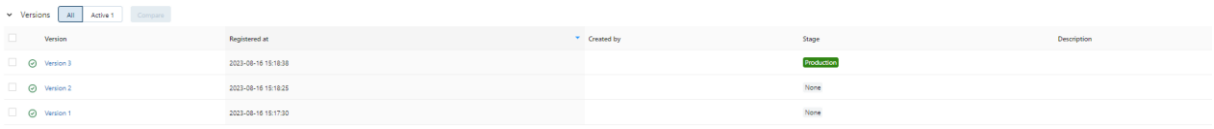

Figure 8: vérifier l'état du modèle

API en Production Pointant vers MLflow : Lorsqu'un modèle a été mis en production, l'API FastAPI a automatiquement pointé vers le modèle correspondant dans MLflow, assurant ainsi la cohérence entre l'entraînement et le déploiement.

Figure 9: Tester le modèle mis en production

Figure 10: Résultat obtenu par l'api

La figure 10 montre que le commentaire saisi dans la figure 9 est classé dans la classe « Pb de suivi /Délai »

Cette architecture complète crée un processus fluide et automatisé pour le déploiement, la gestion et la mise à jour de modèles de machine learning. Elle garantit la continuité des performances tout en optimisant les ressources.

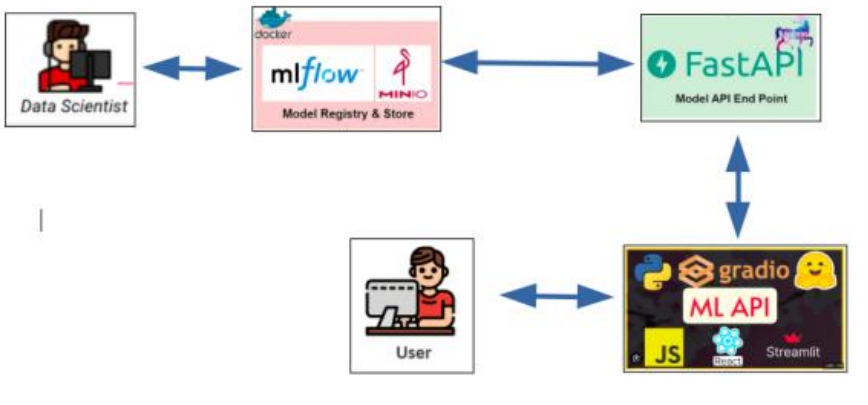

Figure 11: Architecture de la méthodologie MLops

2.3.4 Développement des outils MLOps

Pour faciliter l'installation et la configuration des outils MLOps, j'ai opté pour l'utilisation de Docker Compose, une solution de déploiement basée sur des conteneurs, permettant de créer un environnement isolé et cohérent pour les différentes composantes du système.

2.3.4.1 Installation des outils par Docker Compose

J'ai créé un fichier de configuration Docker Compose décrivant les services nécessaires pour le projet, notamment MLflow, Minio, FastAPI, et BentoML. Chaque service était configuré avec ses dépendances logicielles spécifiques et les paramètres appropriés pour une intégration harmonieuse.

L'utilisation de Docker Compose a simplifié l'installation des outils pour l'équipe. En exécutant une seule commande, tous les services ont été déployés et configurés de manière cohérente, garantissant une mise en place rapide et efficace de l'environnement MLOps.

2.3.4.2 Configuration de MLflow pour stocker les artefacts dans Minio

Une partie cruciale de la méthodologie MLOps est la gestion et le stockage des artefacts générés lors de l'entraînement des modèles. J'ai configuré MLflow pour stocker ces artefacts dans Minio, un service de stockage d'objets, afin de garantir leur persistance et leur accessibilité.

La configuration de MLflow pour utiliser Minio s'est faite en spécifiant les paramètres d'accès à Minio dans le fichier de configuration de MLflow. Ainsi, les artefacts tels que les modèles entraînés, les métriques de performance, les hyperparamètres, etc., étaient sauvegardés de manière sécurisée dans Minio, prêts à être réutilisés ultérieurement.

2.3.4.3 Intégration avec Mlflow et FastAPI

FastAPI a été connecté à MLflow [2] pour créer une API conviviale permettant de servir les modèles déployés. Ainsi, une fois qu'un modèle était mis en production via MLflow, l'API FastAPI pointait automatiquement vers ce modèle, facilitant ainsi son utilisation par les utilisateurs finaux,en utilisant les lignes de codes suivantes :

```
model_uri ="models:/classification/Production"
model=mlflow.pyfunc.load_model(model_uri)
```

2.4 Conclusion

Le développement des outils MLOps lors de ce projet a été essentiel pour faciliter le déploiement des modèles de machine learning dans la ville de Paris. L'utilisation de Docker Compose a permis une installation rapide et cohérente des outils, créant un environnement propice au développement et à la gestion des modèles.

La configuration de MLflow pour stocker les artefacts dans Minio a assuré la sauvegarde et la persistance des informations cruciales pour le suivi et la réutilisation des modèles entraînés, tandis que FastAPI a offert une interface conviviale pour servir les modèles déployés.

En conclusion, le développement et l'intégration des outils MLOps ont été des étapes essentielles dans la mise en place réussie de la méthodologie MLOps pour le déploiement des modèles de machine learning. Ces outils fournissent une base solide pour optimiser le cycle de vie des modèles et faciliter leur utilisation en production dès que les ressources matérielles suffisantes seront disponibles.