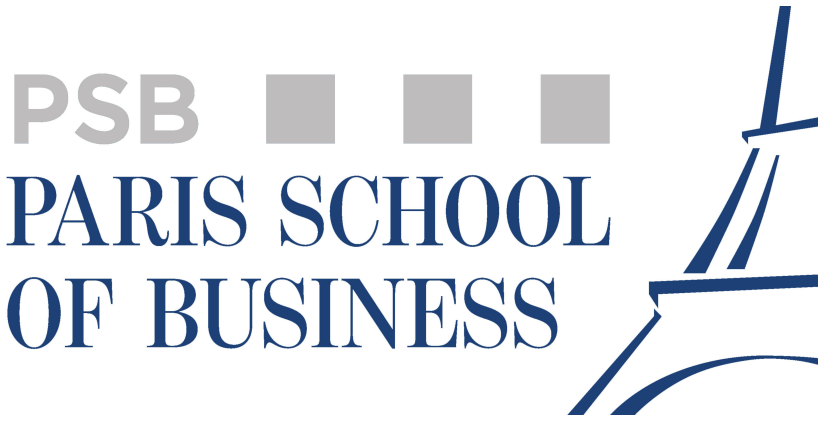
PSB Paris School of Business



MLOps

Zakaria RIDADARAJAT

24 janvier 2021

*

Table des matières

1 Introduction

Le développement des écosystème liés à la data science sont devenu extrêmement riche et dynamique autour des thématiques du machine Learning et l'intelligence artificielle.

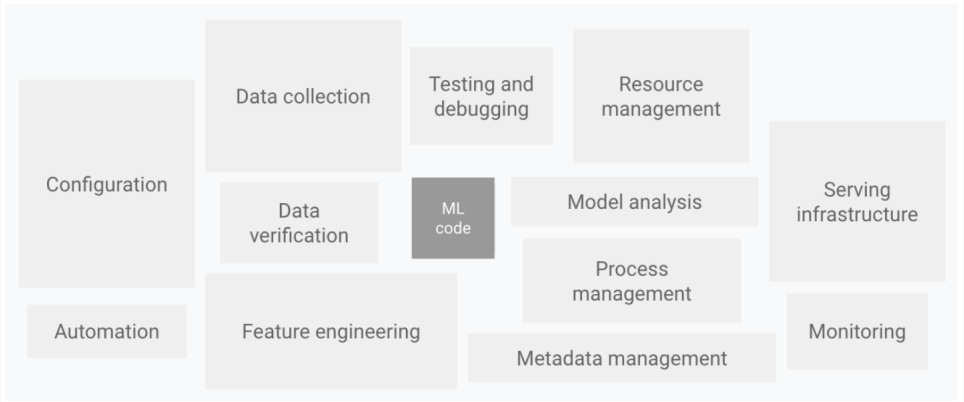
Cet engouement autour du machine Learning a poussé les entreprises à la multiplication de plusieurs projets sur divers domaines. Des projets pour lesquels les outils et les méthodes ne sont pas encore très matures. En l'occurrence, parmi ces problématiques, on retrouve la mise en production des modèles, l'intégration continue (CI), la livraison continue (CD) et les procédures des tests unitaire (Macro-test et Micro-test). Pour répondre à ces problématiques, de nouvelles pratiques regroupées sous le nom de MLOps ont vu le jour afin d'exploiter efficacement les modèles ML.

2 DevOps et MLOps

La culture du MLOps s'inspire des principes du DevOps et les applique à des modèles d'apprentissage automatique à la place de logiciels, unissant les cycles de développement suivis par les data scientist et les machine Learning engineering avec ceux des opérationnel team afin d'assurer la livraison continue des modèles ML avec de bonnes performances. On constate alors que le véritable challenge liée à ces pratiques n'est pas de créer un modèle de ML, mais de créer un système de ML intégré et de le faire fonctionner en production de manière continue.

Mais avant de rentrer sur les détails liés aux caractéristiques du MLOps, nous devons tout d'abord connaitre qu'un système ML intégré ne se résume pas dans le code ML (script) mais plutôt qu'il assembler plusieurs composantes principales dont ce dernier.

Ci-dessous-sous un schéma exhaustif des composantes d'un système ML intégré :

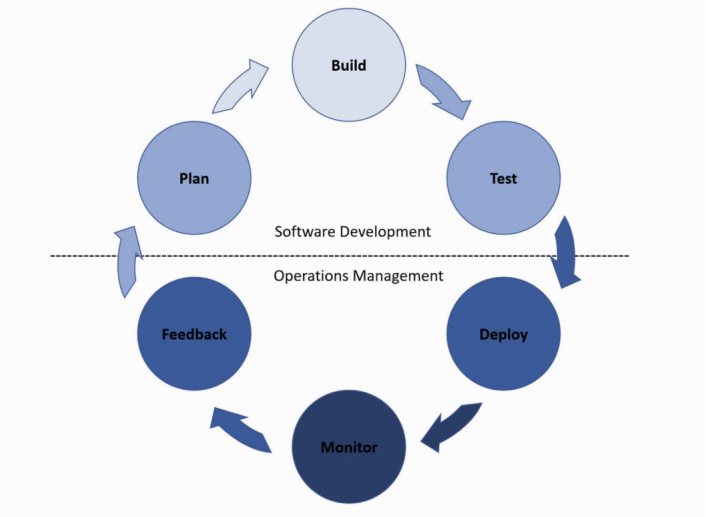


– figure 1 :Composantes d'un système ML intégré

Dans ce schéma, on constate qu'un système ML intégré englobe la configuration des environnements de travail, l'automatisation, la collecte de données, la vérification des données, les tests et le débogage, la gestion des ressources, l'analyse des modèles, la gestion des processus et des métadonnées, l'infrastructure de diffusion et finalement le monitoring.

Maintenant qu'on connait les principales composantes d'un système ML, on passe à la phase qui va mettre en place tous ces composantes citées auparavant et on va les organiser et encadrer par les pratiques du DevOps sous forme d'un workflow.

On illustre ce flux de travail DevOps par le graphe suivant :



– figure 2 :Le flux de travail dans un environnement DevOps

On aperçoit d'après le graphe ci-dessus, que les équipe software développement adopte une méthodologie agile pour le développement des logiciels est qui est résume à travers les étapes de planification, de construction et de test. D'autre part, Les équipes opérationnelles ont la charge du déploiement, de la maintenance et de la collecte des retours sous forme de bugs et de retours utilisateurs et de relayer ces informations aux équipes de développement.

Le MLOps opte la même étape cite auparavant, et cela en considérant le système ml comme un logiciel. La seule différence entre les deux concepts (DevOps et MLOps) réside dans le processus de développement du système ml qui est désigné comme une phase expérimentale.

La phase expérimentale se compose de plusieurs blocs :

— **Extraction de données** : vous sélectionnez et intégrez les données pertinentes de différentes sources de données pour la tâche de ML.

— **Analyse des données** : vous effectuez une analyse exploratoire des données pour comprendre celles qui sont disponibles pour la création du modèle de ML. Ce processus conduit aux étapes suivantes :

1. Comprendre le schéma de données et les caractéristiques attendues par le modèle.
2. Identification des données et l'extraction des caractéristiques nécessaires au modèle.

— **Préparation des données** : les données sont préparées pour la tâche de ML. Cette préparation implique le nettoyage des données, qui consiste à les scinder en trois parties : ensemble d'entraînement, ensemble de validation et ensemble de test. Elle implique également des transformations de données et l'extraction de caractéristiques appliquées au modèle qui résout la tâche cible. Cette étape permet d'obtenir des divisions de données au format préparé.

— **Entraînement de modèles** : le data scientist met en œuvre différents algorithmes avec les données préparées pour entraîner différents modèles de ML. De plus, les algorithmes mis en œuvre font l'objet de réglages d'hyperparamètres afin de produire le modèle de ML le plus performant. Cette étape permet d'obtenir un modèle entraîné.

— **Évaluation du modèle** : le modèle est évalué sur un ensemble de test de données exclues pour évaluer la qualité du modèle. Le résultat de cette étape est un ensemble de métriques permettant d'évaluer la qualité du modèle.

— **Validation du modèle** : le modèle est adapté au déploiement, car ses performances prédictives sont supérieures à une certaine référence.

— Diffusion du modèle : le modèle validé est déployé dans un environnement cible pour réaliser des prédictions. Ce déploiement peut être l'un des suivants :

1. Microservices avec une API REST pour réaliser des prédictions en ligne
2. Modèle intégré à un appareil de périphérie ou mobile
3. Partie d'un système de prédiction par lot

— **Surveillance du modèle** : les performances prédictives du modèle sont surveillées dans la perspective d'appeler une nouvelle itération dans le processus de ML.

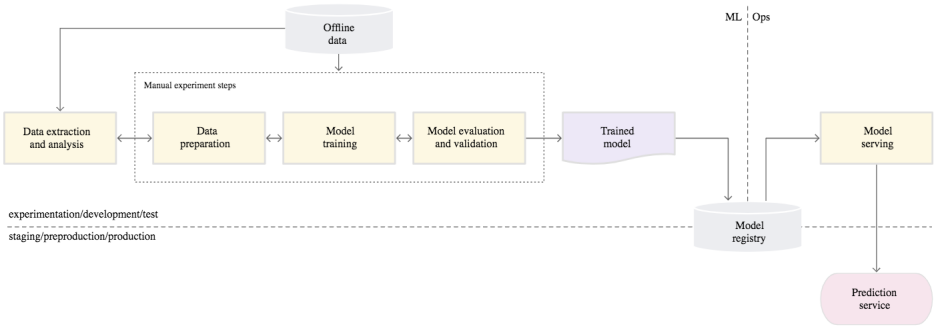
La section suivante décrit trois niveaux de MLOps, du plus courant, qui n'implique aucune automatisation, au niveau d'automatisation complète des pipelines ML et CI/CD.

3 3 niveaux du MLOps

3.1 MLOps de niveau 0 : processus manuel

Un grand nombre d'équipes disposent de data scientists et de chercheurs en ML capables de créer des modèles de pointe, mais leur processus de création et de déploiement de modèles de ML est entièrement manuel. Il s'agit du niveau de maturité de base, ou niveau 0. Le schéma suivant illustre le workflow de ce processus.

Voici les caractéristiques du processus du MLOps au niveau 0 illustre dans le figure ci-dessus :



– figure 3 :Niveau manuel du MLOps

— **Processus manuel, interactif et s'appuyant sur des scripts** : chaque étape est manuelle, y compris l'analyse des données, la préparation des données, l'entraînement du modèle et la validation. Il nécessite d'exécuter à la main chaque étape, et de passer manuellement de l'une à l'autre. Ce processus s'appuie généralement sur un code expérimental écrit et exécuté dans des notebooks de manière interactive par un data scientist, jusqu'à l'obtention d'un modèle fonctionnel.

— **Dissociation entre le ML et les opérations** : le processus fait la distinction entre les data scientists, qui créent le modèle, et les ingénieurs qui le diffusent en tant que service de prédiction. Les data scientists transmettent un modèle entraîné sous forme d'artefact à l'équipe d'ingénieurs chargés de le déployer sur leur infrastructure d'API. Ce transfert peut consister à placer le modèle entraîné dans un emplacement de stockage, à vérifier l'objet du modèle dans un dépôt de code ou à l'importer dans un registre de modèles. Ensuite, les ingénieurs qui déploient le modèle doivent rendre disponibles en production les fonctionnalités requises pour une inférence à faible latence, ce qui peut entraîner un écart entraînement/inférence.

— **Itérations de version peu fréquentes** : le processus suppose que l'équipe de science des données gère quelques modèles qui ne changent pas fréquemment (modifications de la mise en œuvre ou réentraînement du modèle avec de nouvelles données). Une nouvelle version de modèle n'est déployée que quelques fois par an.

— **Pas d'intégration continue** : étant donné le peu de modifications de mise en œuvre attendues, la CI est ignorée. Généralement, le test du code fait partie des notebooks ou de l'exécution des scripts. Les scripts et les notebooks qui mettent en œuvre les étapes de test sont contrôlés par la source et produisent des artefacts tels que des modèles entraînés, des métriques d'évaluation et des visualisations.

— **Pas de livraison continue** : étant donné qu'il n'y a pas de déploiements fréquents de versions de modèle, la CD n'est pas prise en compte.

— **Le déploiement référence le service de prédiction** : le processus ne concerne que le déploiement du modèle entraîné en tant que service de prédiction (par exemple, un micro-service avec une API REST), plutôt que le déploiement de l'ensemble du système de ML.

— **Absence de surveillance active des performances** : le processus ne permet pas de suivre ni de consigner les prédictions et les actions du modèle, qui sont

nécessaires pour détecter la baisse des performances du modèle et d'autres dérives comportementales.

3.2 Difficultés liées au niveau 0

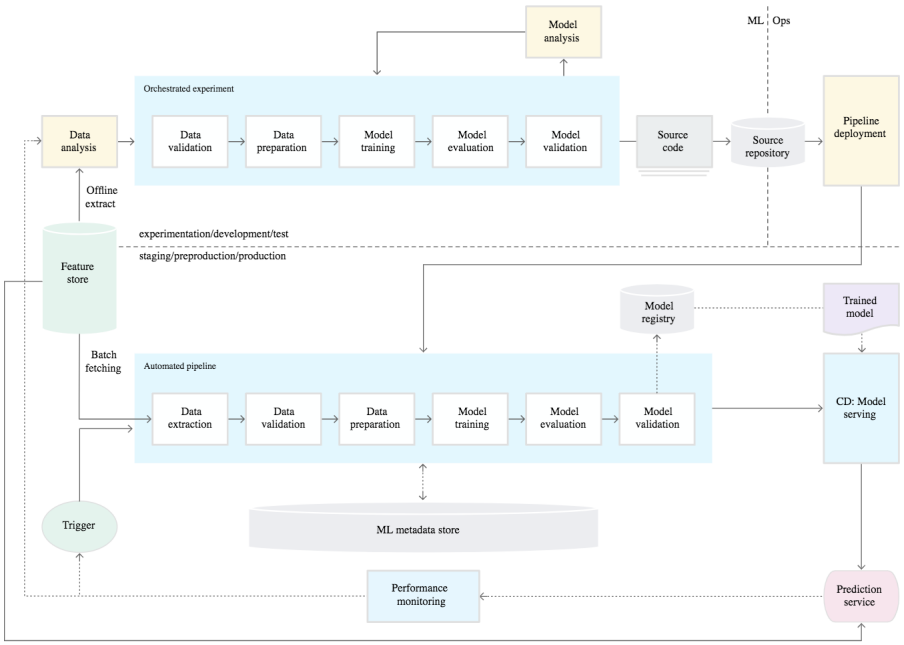
Le MLOps de niveau 0 est couramment utilisé par de nombreuses entreprises qui commencent à appliquer le ML à leurs cas d'utilisation. Ce processus manuel, piloté par les data scientists, peut être suffisant lorsque les modèles sont rarement modifiés ou entraînés. Toutefois, dans la pratique, les modèles présentent souvent des défaillances lorsqu'ils sont déployés en conditions réelles. Ils ne parviennent pas à s'adapter aux changements intervenant au niveau des dynamiques de l'environnement, ou des données décrivant cet environnement.

Pour relever les défis de ce processus manuel, les pratiques MLOps de CI/CD et de CT sont utiles. En déployant un pipeline d'entraînement de ML, vous pouvez activer l'entraînement continu et configurer un système CI/CD pour tester, créer et déployer rapidement de nouvelles mises en œuvre du pipeline de ML. Ces fonctionnalités sont décrites plus en détail dans la section suivante.

3.3 MLOps de niveau 1 : automatisation du pipeline de ML

L'objectif du niveau 1 est d'effectuer un entraînement continu du modèle en automatisant le pipeline de ML, ce qui vous permet d'assurer la livraison continue du service de prédiction de modèle. Pour automatiser le processus d'utilisation de nouvelles données afin de réentraîner des modèles en production, vous devez intégrer au pipeline des étapes automatisées de validation des données et des modèles, ainsi que des déclencheurs de pipeline et une gestion des métadonnées.

La figure suivante est une représentation schématique d'une pipeline de ML automatisé pour l'entraînement continu.



– figure 4 :Niveau 1 du MLOps- Automatisation du pipeline de ML

La liste suivante met en évidence les caractéristiques de la configuration MLOps de niveau 1, comme illustré à la figure 4 :

— **Test rapide** : les étapes de tests de ML sont orchestrées. La transition entre les étapes est automatisée, ce qui permet une itération rapide des tests et une meilleure aptitude à déplacer l'ensemble du pipeline en production.

— **Entraînement continu du modèle en production** : le modèle est automatiquement entraîné en production à l'aide de nouvelles données sur la base de déclencheurs de pipeline en direct, qui sont abordés dans la section suivante.

— **Symétrie expérimentale et opérationnelle** : la mise en œuvre du pipeline exploité dans l'environnement de développement ou de test est appliquée à l'environnement de préproduction et de production, et constitue un aspect essentiel de la pratique MLOps d'unification du DevOps.

— **Code modulaire pour les composants et les pipelines** : pour créer des pipelines de ML, les composants doivent être réutilisables, composables et potentiellement partageables entre les pipelines de ML.

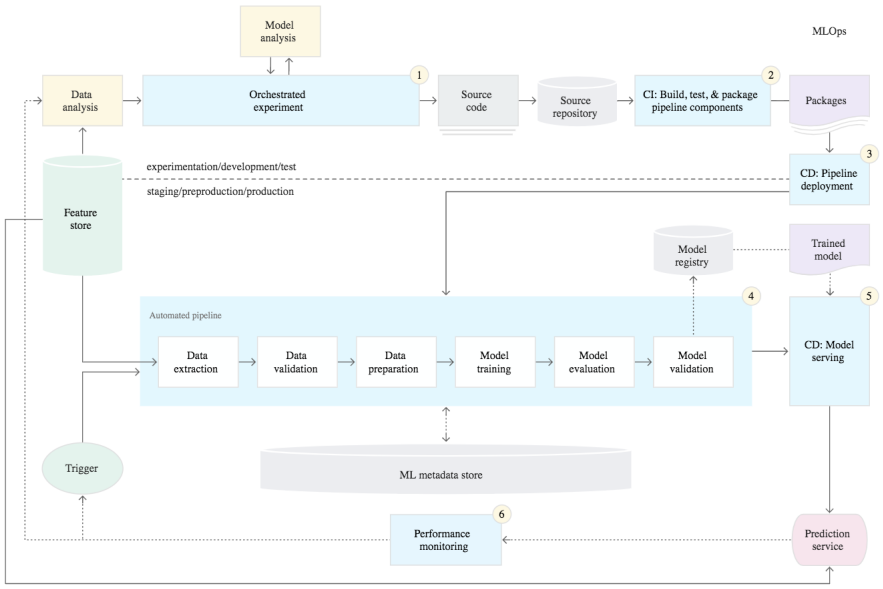
— **Livraison continue de modèles** : un pipeline de ML en production fournit en continu des services de prédiction à de nouveaux modèles entraînés sur de nouvelles données. L'étape de déploiement du modèle, qui diffuse le modèle entraîné et validé en tant que service de prédiction pour les prédictions en ligne, est automatisée.

— **Déploiement du pipeline** : au niveau 0, vous déployez un modèle entraîné en tant que service de prédiction en production. Au niveau 1, vous déployez un pipeline d'entraînement complet, qui s'exécute automatiquement et de manière récurrente pour diffuser le modèle entraîné en tant que service de prédiction.

3.4 MLOps de niveau 2 : automatisation du pipeline CI/CD

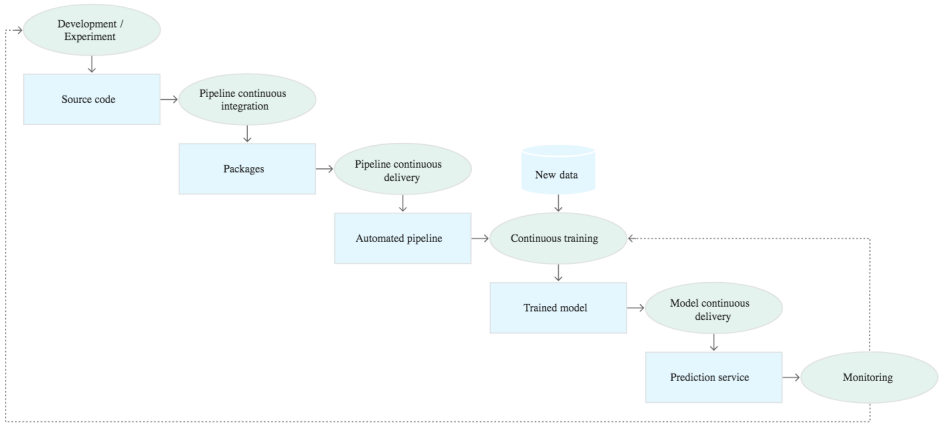
Pour une mise à jour rapide et fiable des pipelines en production, vous avez besoin d'un système CI/CD robuste et automatisé. Ce système CI/CD automatisé permet à vos data scientists d'explorer rapidement de nouvelles idées sur l'extraction de caractéristiques, l'architecture de modèle et les hyperparamètres. Ils peuvent mettre en œuvre ces idées, puis créer, tester et déployer automatiquement les nouveaux composants du pipeline dans l'environnement cible.

Le schéma suivant montre la mise en œuvre du pipeline de ML avec intégration CI/CD, qui présente les caractéristiques de la configuration des pipelines de ML automatisés et les routines CI/CD automatisées.



– figure 5 :Niveau 2 du MLOps- Automatisation du pipeline CI/CD

Caractéristiques Le schéma suivant montre les étapes du pipeline de ML avec routines CI/CD automatisées :



– figure 6 :Niveau 1 du MLOps- Automatisation du pipeline de ML

Le pipeline comprend les étapes suivantes :

1. Développement et expérimentation : vous orchestrez les étapes de test et testez de manière itérative de nouveaux algorithmes de ML et une nouvelle modélisation. Le résultat de cette étape est le code source des étapes du pipeline de ML qui sont ensuite transférées vers un dépôt source.
2. Intégration continue du pipeline : vous créez le code source et exécutez divers tests. Les résultats de cette étape sont les composants du pipeline (packages, exécutables et artefacts) qui seront déployés ultérieurement.
3. Livraison continue du pipeline : vous déployez les artefacts produits par l'étape CI dans l'environnement cible. Le résultat de cette étape est un pipeline déployé avec la nouvelle mise en œuvre du modèle.
4. Déclenchement automatisé : le pipeline est automatiquement exécuté en production en fonction d'un programme ou en réponse à un déclencheur. Le résultat de cette étape est un modèle entraîné qui est transmis au registre de modèles.
5. Livraison continue du modèle : vous diffusez le modèle entraîné en tant que service pour les prédictions. Le résultat de cette étape est un service de prédiction de modèle déployé.
6. Surveillance : vous collectez des statistiques sur les performances du modèle en fonction des données en ligne. Le résultat de cette étape est un déclencheur permettant d'exécuter le pipeline ou d'exécuter un nouveau cycle de tests.

3.5 Intégration continue CI

Dans cette configuration, le pipeline et ses composants sont créés, testés et empaquetés lorsqu'un nouveau code est validé ou transféré vers le dépôt de code source.

En plus de créer des packages, des images de conteneurs et des exécutables, le processus CI peut inclure les tests suivants :

— Tests unitaires de votre logique d'extraction de caractéristiques.
— Tests unitaires des différentes méthodes mises en œuvre dans votre modèle. Par exemple, vous avez une fonction qui accepte une colonne de données catégorielles et vous l'encodez en tant que fonctionnalité one-hot.
— Tests du point de convergence de l'entraînement de votre modèle (surapprentissage de quelques exemples d'enregistrements après diminution de la perte par itérations).
— Tests visant à vérifier que l'entraînement du modèle ne produit pas de valeurs NaN générées par la division par zéro ou la manipulation de valeurs petites ou grandes.
— Tests visant à vérifier que chaque composant du pipeline produit les artefacts attendus.
— Tests de l'intégration entre les composants du pipeline.

3.6 Livraison continue CD

À ce niveau, votre système fournit en continu de nouvelles mises en œuvres de pipeline à l'environnement cible, qui fournit à son tour des services de prédiction du modèle nouvellement entraîné. Pour une livraison continue rapide et fiable des pipelines et des modèles, vous devez tenir compte des points suivants :

— Vérifier la compatibilité du modèle avec l'infrastructure cible avant de déployer votre modèle. Par exemple, vous devez vérifier que les packages requis par le modèle sont installés dans l'environnement de diffusion et que les ressources de mémoire, de calcul et d'accélérateur sont disponibles.
— Tester le service de prédiction en appelant l'API de service avec les entrées attendues, en vous assurant d'obtenir la réponse prévue. Ce test capture généralement les problèmes qui peuvent survenir lorsque vous mettez à jour la version du modèle et qu'une autre entrée est attendue.
— Tester les performances du service de prédiction, ce qui implique d'effectuer des tests de charge afin de capturer des métriques telles que les requêtes par secondes (RPS) et la latence du modèle.
— Valider les données à des fins de réentraînement ou de prédiction par lot.
— Vérifier que les modèles atteignent les objectifs de performances prédictifs avant leur déploiement.
— Déploiement automatisé dans un environnement de test, par exemple, un déploiement déclenché par l'envoi de code à la branche de développement.
— Déploiement semi-automatisé dans un environnement de préproduction, par exemple, un déploiement déclenché par la fusion du code dans la branche principale après l'approbation des modifications par les évaluateurs.

— Déploiement manuel dans un environnement de production après plusieurs exécutions réussies du pipeline dans l'environnement de préproduction.

Pour résumer, la mise en œuvre du ML dans un environnement de production ne consiste pas seulement à déployer votre modèle en tant qu'API pour la prédiction. Il s'agit aussi de déployer un pipeline de ML capable d'automatiser le réentraînement et le déploiement de nouveaux modèles. La configuration d'un système CI/CD vous permet de tester et de déployer automatiquement de nouvelles mises en œuvre de pipeline. Avec un tel système, on peut faire face à l'évolution rapide des données et des environnement de travail.

Références

[1] Beginning MLOps with MLFlow :"Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure"
Auteur : Sridhar Alla.