Projet 5
Catégorisez automatiquement des questions

OPENCLASSROOMS

Août 2024

Note Technique

Approche et Outils pour la Généralisation de l'Approche MLOps

Présenté par

Adrien Claire

TABLE DES MATIÈRES

À PROPOS

But du Projet :

Le projet 5, intitulé "Catégorisation Automatique des Questions", a pour objectif de développer un modèle de classification supervisée capable d'attribuer automatiquement des tags pertinents à chaque question posée sur une plateforme comme Stack Overflow. Ces tags jouent un rôle essentiel en aidant les utilisateurs à trouver des questions similaires, à regrouper les sujets connexes et à améliorer la navigation sur le site. Le défi consiste à créer un modèle qui non seulement comprend le contenu des questions, mais qui est également capable de prédire les tags les plus appropriés avec un haut degré de précision.

Mise en Œuvre :

Pour répondre à ce besoin, une approche de Machine Learning a été adoptée, utilisant des techniques avancées telles que l'ingénierie des caractéristiques, le traitement du langage naturel (NLP) et des algorithmes de classification. Le projet a été réalisé en plusieurs étapes clés :

Exploration et Pré-traitement des Données :

- Une analyse approfondie des données textuelles a été menée, comprenant des analyses univariées et multivariées pour comprendre la distribution des tags et des termes.
- Le pré-traitement des données a inclus des étapes de nettoyage, de tokenization, et de réduction de la dimensionnalité, essentielles pour améliorer la qualité des données d'entrée du modèle.

Feature Engineering et Modélisation :

- Des techniques telles que le Bag of Words, TF-IDF et Word2Vec ont été utilisées pour transformer les questions en vecteurs numériques exploitables par les algorithmes de machine learning.
- Plusieurs modèles de classification ont été testés, dont la Régression Logistique, les Machines à Vecteurs de Support (SVM), et les modèles basés sur des réseaux neuronaux profonds. Ces modèles ont été évalués sur des métriques telles que la F1-score et le Jaccard index pour sélectionner le modèle le plus performant.

Déploiement :

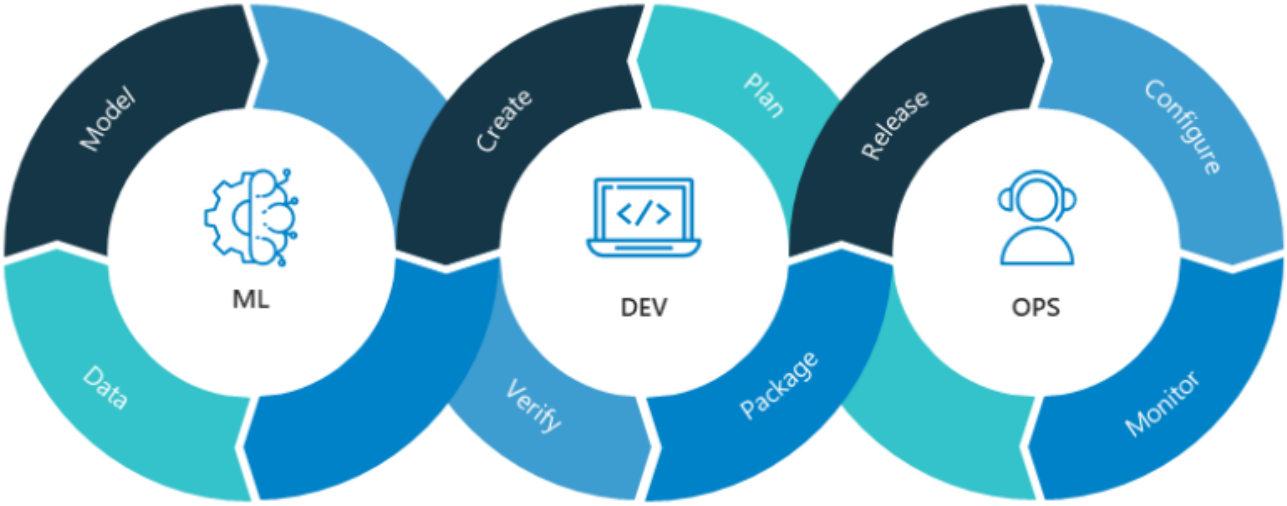
Le modèle final a été intégré dans une API de prédiction de mots clés, déployée sur le Cloud, et capable de traiter les requêtes en temps réel.

Conclusion :

Ce projet illustre la puissance des techniques de Machine Learning pour résoudre des problèmes complexes de catégorisation.
Cette note technique démontrera l'importance des bonnes pratiques de MLOps pour garantir la robustesse et la stabilité du modèle sur le long terme. La solution développée sera non seulement capable d'améliorer l'expérience utilisateur sur des plateformes comme Stack Overflow, mais elle offre également un cadre réutilisable pour d'autres projets de classification dans divers domaines.

Introduction à MLOps

DÉFINITION ET IMPORTANCE

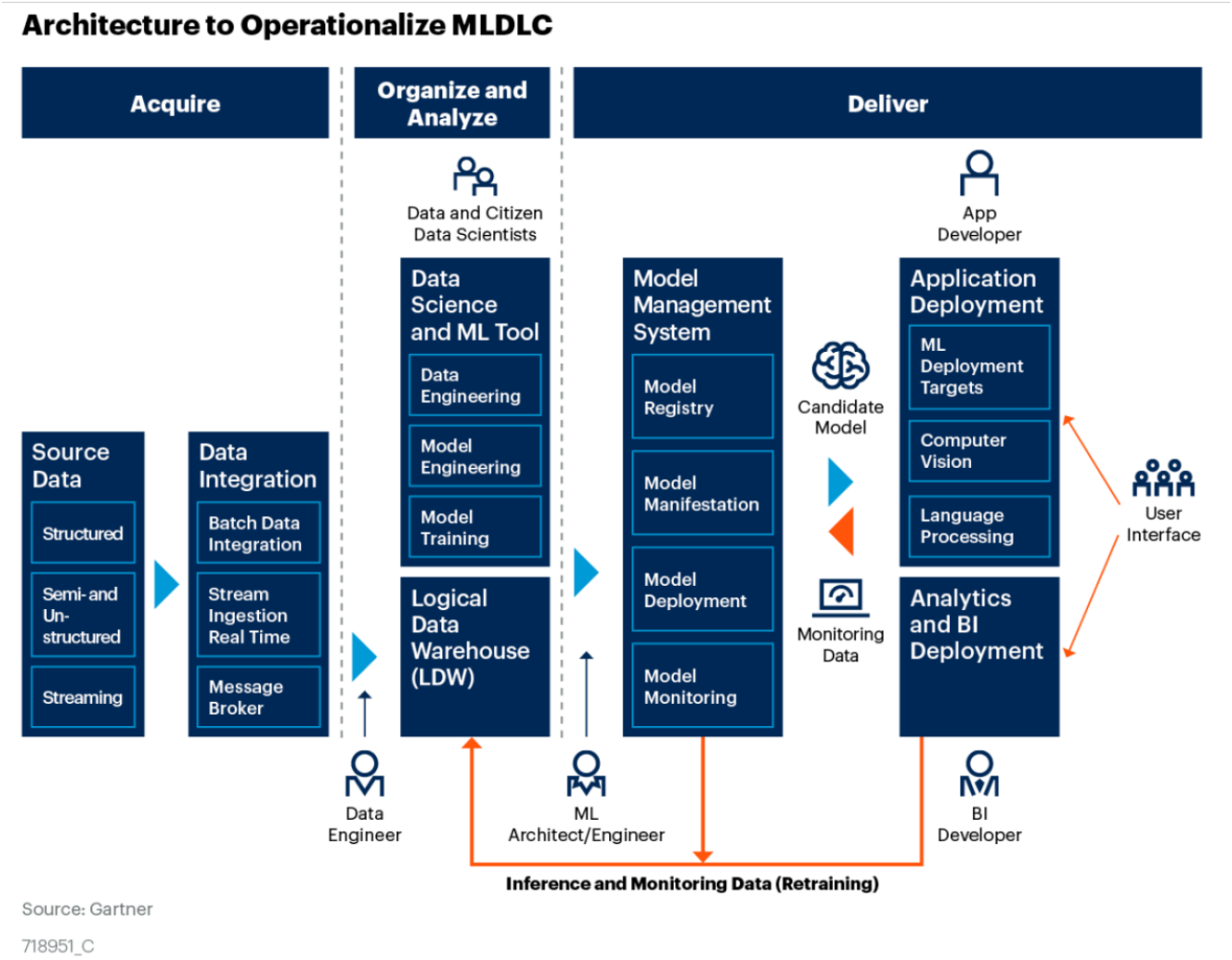


MLOps, ou Machine Learning Operations, désigne un ensemble de pratiques qui fusionnent le Machine Learning (ML) avec les principes d'ingénierie logicielle pour créer une infrastructure permettant le déploiement, la surveillance et la mise à jour continue des modèles ML. L'objectif principal de MLOps est d'assurer la fiabilité, l'évolutivité et la reproductibilité des modèles en production, tout en gérant les défis spécifiques tels que la dérive des données (data drift) et la dérive conceptuelle (concept drift).

Challenges Résolus par MLOps

MLOps résout plusieurs défis majeurs rencontrés dans les projets ML :

- **Versioning des Modèles** : Assurer la traçabilité et la gestion des différentes versions des modèles.
- **Scalabilité** : Déployer et maintenir des modèles à grande échelle dans des environnements de production.
- **Reproductibilité** : Garantir que les résultats peuvent être reproduits dans différents environnements, ce qui est essentiel pour la validation scientifique et les audits.
- **Détection de Dérive** : Surveiller les modèles en production pour identifier les dérives qui peuvent nuire à leur performance, comme le "data drift" et le "concept drift".



Source: Gartner

718951_C

Mise en Œuvre des Pipelines

Les pipelines automatisent l'ensemble du processus ML, de la collecte des données au déploiement du modèle, garantissant ainsi la reproductibilité, la traçabilité et l'évolutivité de chaque étape. En utilisant des pipelines bien définis, il devient plus facile d'intégrer des données nouvelles, de mettre à jour les modèles, et de s'assurer que toutes les étapes sont exécutées de manière cohérente et contrôlée.

ÉTAPES ET OUTILS POUR LA GESTION DES PIPELINES

Recherche et Extraction des Données

- **Description** : La première étape du pipeline consiste à identifier, accéder et extraire les données nécessaires à la modélisation. Cette étape est cruciale pour garantir que les données disponibles sont à jour et pertinentes pour les objectifs du projet.

- **Outils:**
    - **AWS Glue** : AWS Glue est un service d'extraction, transformation et chargement (ETL) entièrement géré. Il permet de cataloguer les données, de les nettoyer et de les transformer à partir de sources multiples (bases de données, fichiers plats, API, etc.). AWS Glue peut automatiser la recherche et l'extraction des données, en les rendant disponibles pour les étapes suivantes du pipeline.

- **Cas d'utilisation** : AWS Glue est utilisé pour extraire des données de divers formats et sources, les transformer de manière cohérente, et les rendre prêtes pour le traitement et l'analyse.

Nettoyage et Préparation des Données

- **Description** : Une fois les données extraites, elles doivent être nettoyées et préparées pour l'analyse. Cela inclut la gestion des valeurs manquantes, la normalisation des formats, et la transformation des données brutes en caractéristiques exploitables (features).
- **Outils:**
    - **AWS Glue** : En plus de l'extraction, AWS Glue permet également de nettoyer les données à grande échelle.
    - **Kedro** : Kedro est un framework qui aide à structurer le code de traitement des données, rendant le pipeline plus maintenable et reproductible. Il est particulièrement utile pour les projets complexes nécessitant une gestion rigoureuse des transformations de données.
- **Cas d'utilisation** : Les outils comme AWS Glue et Kedro automatisent le nettoyage et la transformation des données, ce qui est essentiel pour garantir des résultats de modélisation de haute qualité.

Feature Engineering

- **Description** : Cette étape consiste à transformer les données brutes en caractéristiques (features) qui seront utilisées par le modèle de machine learning. Cela inclut l'encodage de variables catégorielles, la création de nouvelles caractéristiques, et la réduction de dimension.
- Outils:
    - **Scikit-learn Pipelines** : Un framework populaire pour automatiser les transformations de données en features.
    - **Kedro** : Utilisé pour structurer le code de transformation des features et intégrer ces étapes dans le pipeline global.
- **Cas d'utilisation** : L'intégration de la création des features dans un pipeline garantit que toutes les données sont traitées de manière uniforme avant l'entraînement du modèle.

Modélisation et Entraînement

- **Description** : Une fois les données préparées, elles sont utilisées pour entraîner un modèle de machine learning. Cette étape peut inclure des expérimentations avec différents algorithmes et hyperparamètres pour identifier le modèle le plus performant.
- Outils:
    - **AWS SageMaker** : SageMaker permet d'entraîner des modèles à grande échelle dans un environnement cloud. Il s'intègre parfaitement avec les données préparées dans AWS Glue et les pipelines de SageMaker, facilitant l'entraînement et le déploiement des modèles.
    - **MLFlow Pipelines** : Permet de gérer le cycle de vie du modèle, du suivi des expérimentations à l'enregistrement des modèles.

Cas d'utilisation : SageMaker est idéal pour les entreprises qui souhaitent entraîner et déployer des modèles ML à grande échelle sur AWS, tandis que MLFlow permet un suivi granulaire des expérimentations.

Évaluation et Validation

- **Description** : Après l'entraînement, le modèle est évalué sur un jeu de données de test pour vérifier qu'il généralise bien. Cette étape est cruciale pour s'assurer que le modèle fonctionne correctement sur des données non vues.
- Outils:
    - **SageMaker Model Monitor** : Permet de surveiller la qualité du modèle en production, en détectant les anomalies et les dérives (voir Annexe pour plus de détails).
    - **MLFlow** : Utilisé pour évaluer et comparer les performances des modèles au fil du temps.
- **Cas d'utilisation** : Ces outils aident à s'assurer que le modèle est performant et reste pertinent après son déploiement.

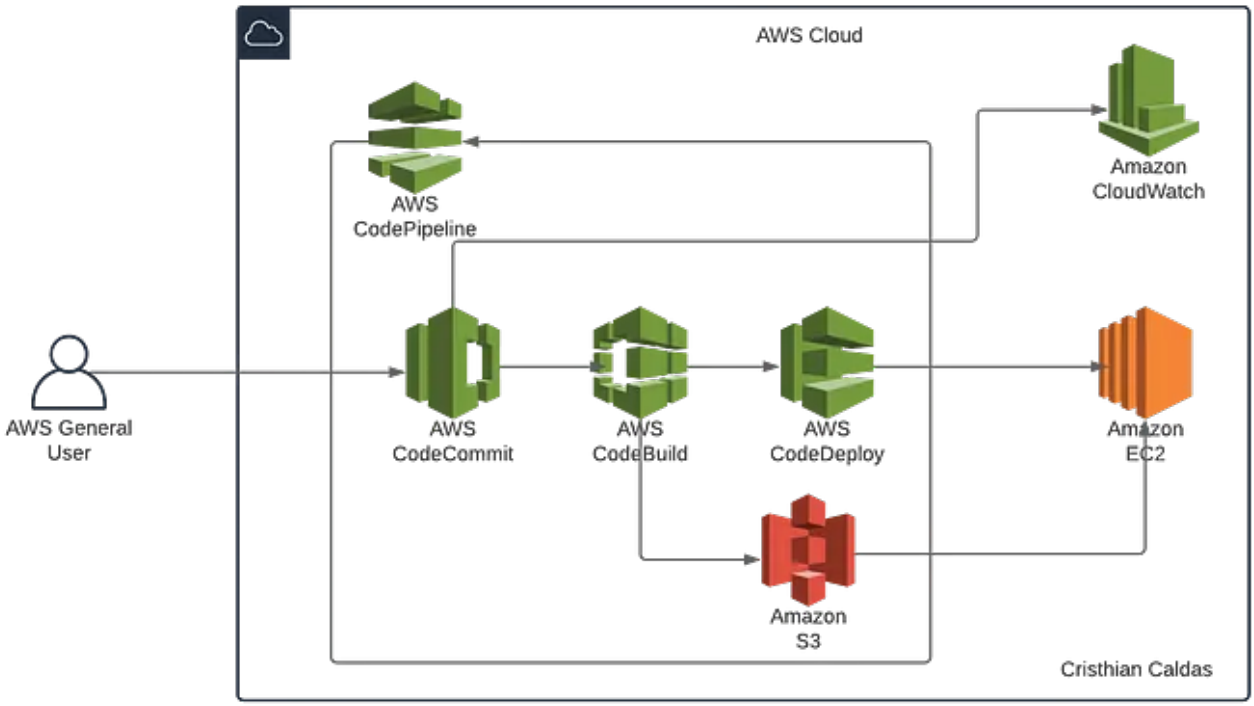
Déploiement et Suivi

- **Description** : Le modèle validé est déployé en production où il est surveillé pour détecter des changements de performance, des dérives de données (data drift) ou des dérives de concept (concept drift). (Voir annexes)

- Outils:
    - **AWS CodePipeline** : Automatiser le déploiement du modèle en utilisant un pipeline CI/CD. CodePipeline gère l'intégration continue et le déploiement continu, en automatisant les tests et le déploiement des modèles après chaque mise à jour.

    - **Prometheus & EvidentlyAI** : Utilisés pour surveiller la performance du modèle en production, détectant toute forme de drift et déclenchant un réentraînement si nécessaire (voir Annexe pour les types de drifts).

- **Cas d'utilisation** : AWS CodePipeline assure que les modèles sont déployés de manière cohérente et rapide, tandis que Prometheus et EvidentlyAI permettent de surveiller et de maintenir la performance du modèle au fil du temps.



Surveillance du Modèle et Suivi des Performances

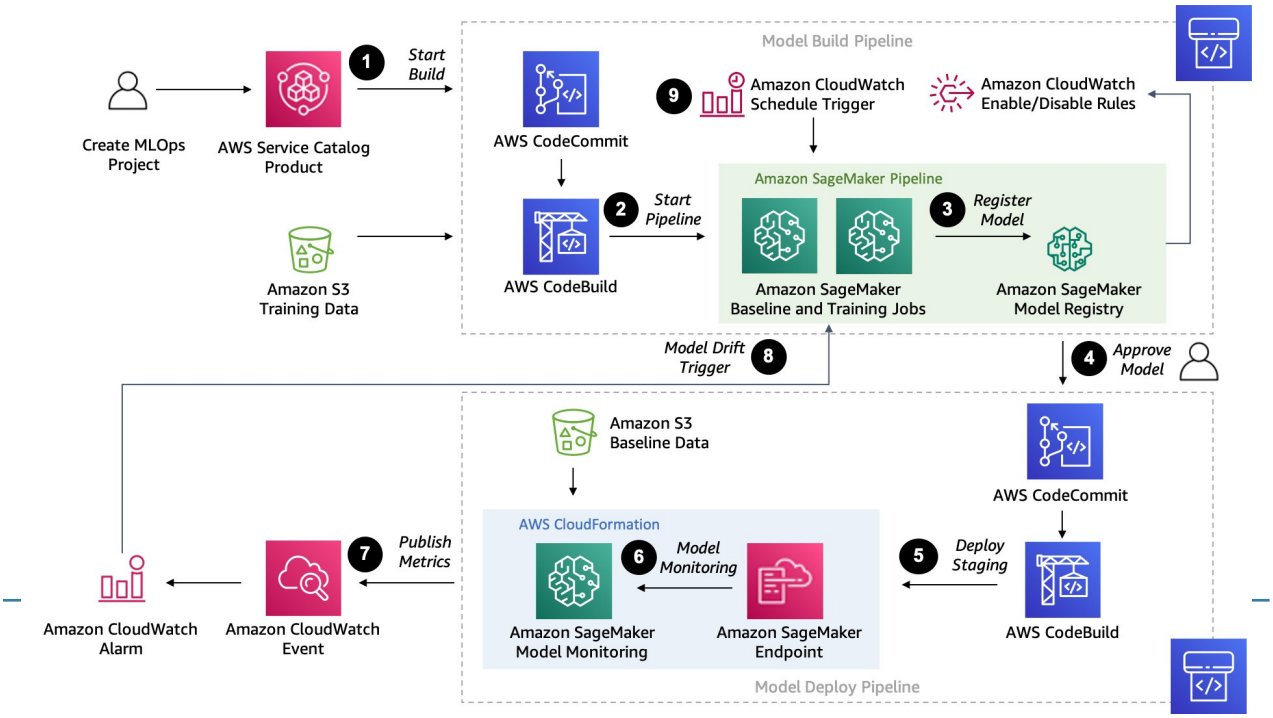
IMPORTANCE DE LA SURVEILLANCE EN PRODUCTION

Une fois les modèles déployés en production, il est crucial de surveiller leur performance pour détecter toute dérive ou perte de précision. Cette surveillance permet de s'assurer que le modèle continue à fournir des prédictions fiables, et de réagir rapidement en cas de besoin de ré-entraînement ou d'ajustement. La mise en place d'alertes est essentielle pour anticiper les dégradations de performance et garantir une maintenance proactive du modèle.

OUTILS DE SURVEILLANCE ET DE SUIVI

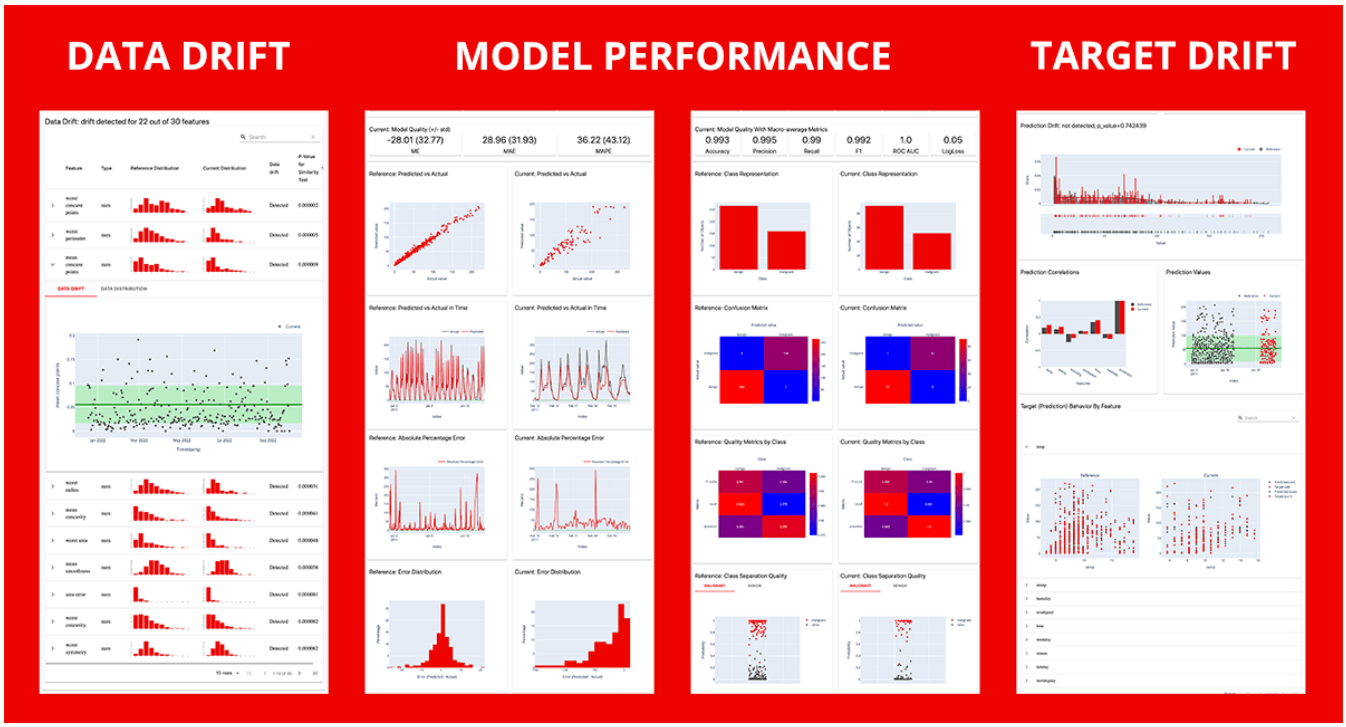
- **AWS Sagemaker Model Monitor**
  - **Description** : SageMaker Model Monitor permet de surveiller en continu la qualité des modèles déployés sur SageMaker, en identifiant les dérives des données en entrée par rapport aux données de formation. Il offre des alertes et des rapports automatiques qui aident les équipes à maintenir la performance des modèles.

  - **Cas d'utilisation** : Parfait pour les environnements AWS où les modèles doivent être surveillés en continu pour garantir leur performance. Lorsqu'une dérive est détectée, une alerte est générée, permettant aux équipes de réagir rapidement pour réentraîner le modèle ou ajuster les paramètres.



- **EvidentlyAI**
  - **Description** : EvidentlyAI fournit des outils de surveillance pour les modèles ML en production, se concentrant sur la détection de la dérive des données et la dégradation des performances. Il génère des rapports interactifs et des alertes basées sur les métriques de performance.

  - **Cas d'utilisation** : Utilisé pour garantir la stabilité des modèles dans le temps, en particulier dans des contextes où les données peuvent évoluer rapidement. EvidentlyAI peut être configuré pour envoyer des alertes lorsque des changements significatifs dans les données ou les performances du modèle sont détectés, permettant ainsi une intervention rapide.



- **Prometheus**
  - **Description** : Prometheus est une plateforme de surveillance open-source qui peut être utilisée pour suivre les performances des modèles ML en production. Il offre une intégration facile avec les systèmes existants pour surveiller les métriques en temps réel et déclencher des alertes.
  - **Cas d'utilisation** : Prometheus est adapté pour la surveillance à grande échelle des systèmes ML, avec une capacité à suivre des métriques en temps réel et à déclencher des alertes lorsque des seuils critiques sont dépassés.

Cas d'Utilisation ou Workflow Exemple

INTÉGRATION DANS LE PROJET

Dans le cadre de votre projet, l'intégration de ces outils pourrait se dérouler de la manière suivante :

- **AWS Sagemaker Pipelines** pour automatiser les étapes de préparation des données, d'entraînement et de déploiement des modèles.
- **MLFlow** pour le suivi des expérimentations et le versioning des modèles.
- **AWS Sagemaker Model Monitor** pour la surveillance continue des modèles en production.
- **EvidentlyAI** pour des rapports détaillés et des alertes sur la performance des modèles.

Exemple de Résultats et de Métriques

Une fois le modèle déployé, le succès pourrait être mesuré à travers des métriques telles que le score F1 et le Jaccard index, avec une surveillance continue pour détecter les dérives et ajuster le modèle en conséquence.

Conclusion et Recommandations

RÉSUMÉ DES AVANTAGES DE MLOPS

Adopter une approche MLOps dans votre projet apporte plusieurs avantages :

- **Fiabilité** : Les modèles sont continuellement suivis et ajustés pour maintenir leur performance.
- **Scalabilité** : Les pipelines automatisés facilitent l'adaptation à des volumes de données croissants.
- **Reproductibilité** : Les workflows définis garantissent que les résultats peuvent être reproduits dans différents environnements.

Commentaires

Il est important de noter qu'il existe un nombre exhaustif d'outils pour implémenter une approche MLOps. Dans ce projet, nous avons choisi de nous concentrer sur les outils AWS, tels que SageMaker et SageMaker Model Monitor, car nous avons déployé notre modèle sur AWS Elastic Beanstalk, ce qui faciliterai l'intégration dans l'écosystème AWS. Toutefois, d'autres outils, comme MLFlow, Prometheus, et EvidentlyAI, peuvent également être utilisés de manière complémentaire pour enrichir le pipeline MLOps.

La mise en place d'alertes via ces outils est cruciale pour assurer une surveillance proactive des modèles en production. Ces alertes permettent de réagir rapidement en cas de dérive, garantissant ainsi que les modèles restent performants et pertinents au fil du temps. Certains de ces outils sont open-source, offrant ainsi une grande flexibilité et une communauté de soutien, tandis que d'autres sont propriétaires, avec des fonctionnalités avancées adaptées à des besoins spécifiques.

Annexes

LES DIFFÉRENTS TYPES DE DRIFTS

DATA DRIFT

- **Description** : Le data drift se produit lorsque la distribution statistique des questions posées sur Stack Overflow change au fil du temps. Par exemple, la popularité de certains langages de programmation ou frameworks pourrait augmenter ou diminuer, modifiant ainsi les types de questions posées.

- **Exemple** : Si un modèle de prédiction de tags a été entraîné lorsque JavaScript était le langage de programmation le plus discuté sur Stack Overflow, et que la popularité de Python augmente avec le temps, les performances du modèle pourraient diminuer si ce changement n'est pas pris en compte. Le modèle pourrait par exemple continuer à prédire des tags liés à JavaScript alors que les questions sont désormais davantage orientées vers Python.

CONCEPT DRIFT

- **Description** : Le concept drift survient lorsque la relation entre le texte des questions et les tags associés change avec le temps. Cela peut se produire lorsque de nouveaux concepts ou frameworks émergent, ou lorsque l'usage de termes spécifiques évolue.

- **Exemple** : Supposons qu'un nouveau framework JavaScript, comme un successeur de React, devienne populaire. Les développeurs pourraient commencer à utiliser de nouveaux termes spécifiques à ce framework. Un modèle formé avant l'émergence de ce framework pourrait ne pas reconnaître ces nouveaux termes et ne pas être capable de prédire les tags corrects, comme le tag correspondant au nouveau framework.

FEATURE DRIFT

- **Description** : Le feature drift se produit lorsque l'importance des caractéristiques textuelles extraites des questions Stack Overflow change avec le temps. Par exemple, certains mots-clés ou phrases dans les questions peuvent devenir plus ou moins pertinents pour prédire les tags associés.

- **Exemple** : Si des changements dans la manière dont les développeurs posent leurs questions se produisent — par exemple, s'ils deviennent plus succincts ou utilisent de nouveaux termes techniques

— les caractéristiques textuelles que le modèle utilise pour prédire les tags pourraient ne plus être aussi pertinentes. Cela pourrait entraîner une diminution de la précision du modèle, comme dans le cas où un mot-clé important pour prédire le tag "machine learning" devient moins courant au profit d'un nouveau terme technique.