

# Chatbot :

## Méthodologie du pilotage de la performance du modèle

Lors de sa mise en production, notre chatbot sera confronté à des demandes utilisateurs qui différeront certainement du jeu de données initial. Ce document a pour but de prévoir une méthode de maintenance pour que le modèle conserve des performance acceptables tout au long de sa durée de vie.

Avant toute chose, il est nécessaire de déterminer la ou les conditions de mise à jour du modèle et par conséquent, une métrique adaptée.

On peut mettre à jour le modèle quand les performances passent en dessous d'un certain seuil en surveillant les alertes générées par le service App Insights (par exemple si une alerte est déclenchée dans plus de 1% des cas).

On peut aussi calculer un score (par exemple le F1-score) sur un jeu de données composé des récentes demandes utilisateur et mettre à jour le modèle si ce score est inférieur à une limite, que ce soit un seuil fixe (0,95) ou un pourcentage du score initial (90%).

Dans les deux cas, une évaluation périodique est à prévoir puisque de nouvelles interactions entre le chatbot et les utilisateurs sont sensées survenir très fréquemment.

Dans la suite de ce document j'utiliserais la première méthode (alertes App Insights) car plus concrète et proche du but, qui est d'offrir à l'utilisateur une expérience pertinente.

En effet, se fier à un score ajoute un niveau d'abstraction supplémentaire et pourrait mener à des situations où, par exemple, une mise à jour est effectuée alors que les utilisateurs sont satisfait de l'état actuel du produit.

La condition de mise à jour étant établie, il suffit d'attendre que la performance du modèle ne respecte plus les standards énoncés précédemment pour qu'il devienne nécessaire de ré-entraîner le modèle avec un jeu de donnée incluant les nouvelles interactions des utilisateurs.

Le processus de mis à jour comporte plusieurs étapes : récolter et formater les nouveaux dialogues, les labelliser pour les inclure dans le jeu d'entraînement, entraîner le modèle, l'évaluer, puis le déployer.

Les outils nécessaires à ces étapes dépendent de la popularité de notre produit et donc de la quantité de données à traiter.

Si il s'agit de petites quantités, il est possible de stocker les interactions, les labelliser et créer le jeu d'entraînement localement.

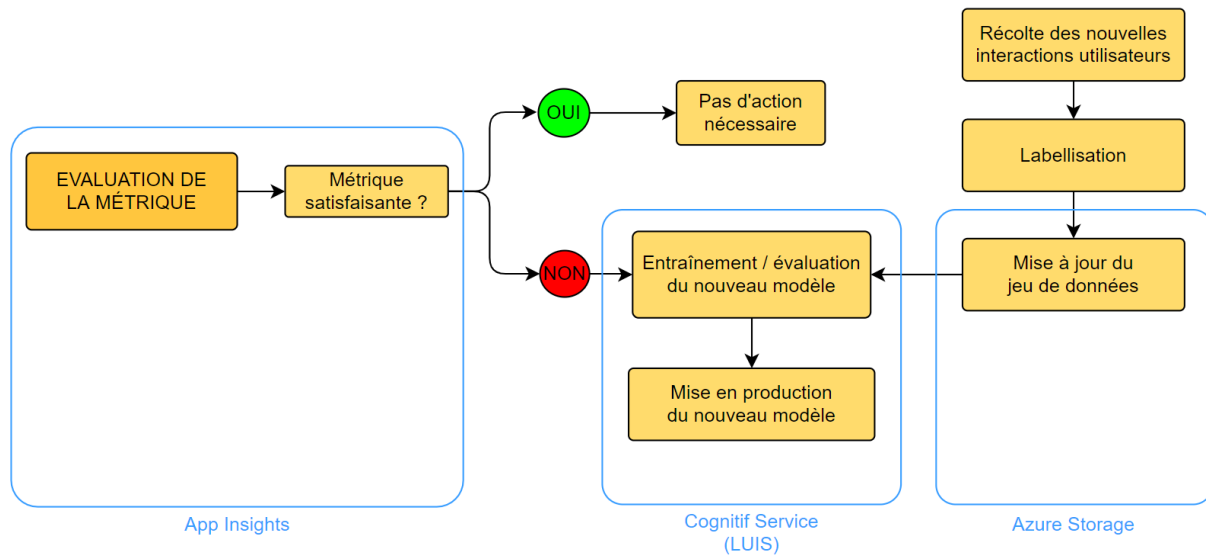
Pour gérer des quantités plus conséquentes, il est possible d'utiliser le service Azure Storage qui permettrait au fil du temps de stocker un jeu de données en constante augmentation.

Quant à la labellisation des nouvelles données, cela pourrait rapidement devenir un problème si cette tâche est effectué par des humains. Il deviendra alors nécessaire de recruter plus de personnel ou de faire appel à une entreprise tiers spécialisée (SmartOne, Clickworker, etc...).

Il serait ainsi possible d'automatiser le processus de récolte et de mise à jour des données, qui n'aurait plus qu'à être consultées lorsqu'une mise à jour du modèle devient nécessaire.

L'entraînement et la mise en production du nouveau modèle est exécuté avec Azure Cognitif Services, il suffit donc de reprendre les scripts utilisés pour la première mise en production du modèle LUIS.

Schéma d'amélioration continue :



Pour garantir la qualité du chatbot, cette méthodologie est à appliquer périodiquement (par exemple une fois par semaine), mais aussi pendant ou avant une forte période d'activité.

En résumé, les alertes de performance App Insights permettent de déclencher un ré-entraînement du modèle LUIS avec un jeu de données contenant les nouvelles interactions utilisateurs (déjà traité et sauvegardé dans Azure Storage).