# Choix de la Technologie pour l’environnement de Dev MLOPS

## Introduction

Deux choix de technologie semble être intéressante, Python ou .Net (Du fait que l’entreprise TEDIES se trouve fortement intégrées dans l’écosystème Microsoft)

## Analyse

### Écosystème et bibliothèques

Python :

* Richesse de l'écosystème : Python est largement utilisé dans le domaine de l'apprentissage automatique (ML) et de l'apprentissage profond (DL). Il possède un large éventail de bibliothèques telles que TensorFlow, PyTorch, scikit-learn, Keras, et Pandas.
* Communauté active : Une communauté large et active qui contribue régulièrement à l'amélioration et à la documentation des outils.
* Outils MLOps spécifiques : Des outils comme MLflow, DVC (Data Version Control), et Kubeflow sont bien intégrés avec Python.

#### .NET :

* Support ML : .NET a développé des bibliothèques comme ML.NET pour intégrer le ML dans les applications .NET ou encore le projet Catalyst. Cependant, ces bibliothèques ne sont pas aussi matures ou répandues que celles de Python.
* Interopérabilité : .NET peut interagir avec des bibliothèques Python via des interopérabilités comme Python.NET, mais cela ajoute de la complexité.

### Performance

Python :

* Flexibilité mais performance variable : Bien que Python soit flexible et facile à utiliser, il n'est pas aussi performant que certains langages compilés pour les tâches intensives en calcul. Cependant, les bibliothèques comme NumPy utilisent des extensions C pour améliorer les performances.

#### .NET :

* Performances optimisées : Les applications .NET, en particulier celles écrites en C#, bénéficient de la performance d'un langage compilé, ce qui peut être un avantage pour certaines tâches de ML intensives en calcul.

### Intégration et déploiement

#### Python :

* Flexibilité de déploiement : Les modèles ML développés en Python peuvent être déployés sur diverses plateformes, y compris les serveurs, les services cloud (AWS, Azure, Google Cloud), et même les appareils mobiles via des frameworks comme TensorFlow Lite.
* Conteneurisation : Bien supporté par des outils comme Docker et Kubernetes, ce qui est crucial pour les environnements MLOps.

#### . NET :

* Écosystème Azure : Fortement intégré avec Microsoft Azure, ce qui peut simplifier le déploiement pour les entreprises utilisant Azure. Cependant, cela peut limiter la flexibilité en matière de choix de plateforme.

## Courbe d'apprentissage et adoption

#### Python :

* Courbe d'apprentissage douce : Python est souvent considéré comme plus facile à apprendre, ce qui peut accélérer le développement et le prototypage.
* Large adoption académique et industrielle : Utilisé massivement par les chercheurs et les industries, facilitant le recrutement et l'accès à des ressources pédagogiques.

#### . NET :

* Courbe d'apprentissage plus raide : Pour les développeurs n'ayant pas d'expérience avec les technologies Microsoft, .NET peut avoir une courbe d'apprentissage plus raide.
* Adoption principalement industrielle : Majoritairement adopté par les entreprises qui ont déjà des écosystèmes .NET en place.

## Support et maintenance

### Python :

* Open Source : La majorité des bibliothèques Python pour le ML sont open source et bénéficient de contributions régulières de la communauté.
* Documentation et forums : Une vaste quantité de documentation et de forums de support disponibles.

#### .NET :

* Support commercial : Supporte par Microsoft, ce qui peut être un avantage pour les entreprises cherchant un support commercial.
* Écosystème de développement solide : Des outils de développement robustes comme Visual Studio, facilitant la gestion des projets et le développement.

## Conclusion

Après une série de tests approfondis et malgré le fait que l’entreprise TEDIES soit fortement intégrée dans l’écosystème Microsoft, nous avons finalement opté pour la solution Python pour notre environnement de développement MLOPS. Bien que .NET présente certains avantages indéniables, notamment en termes de performance et d’intégration avec les outils Microsoft, plusieurs raisons nous ont conduits à privilégier Python.

Tout d’abord, la simplicité de mise en place et de modification offerte par Python constitue un atout majeur pour notre environnement de développement MLOPS. Python est reconnu pour sa courbe d'apprentissage douce, sa flexibilité et sa richesse en bibliothèques dédiées au machine learning et à l’intelligence artificielle. Cette facilité d'utilisation nous permet de gagner en efficacité et en rapidité lors du développement, des tests et du réentrainement des modèles.

De plus, l’utilisation de Python garantit que les tests soient effectués dans des conditions identiques à celles de la création des modèles. Cette cohérence entre l'environnement de développement et de test est essentielle pour assurer la fiabilité et la reproductibilité des résultats. Les outils et bibliothèques Python, tels que TensorFlow, PyTorch, et scikit-learn, offrent une vaste gamme de fonctionnalités qui simplifient le processus de création, de formation et d'optimisation des modèles.