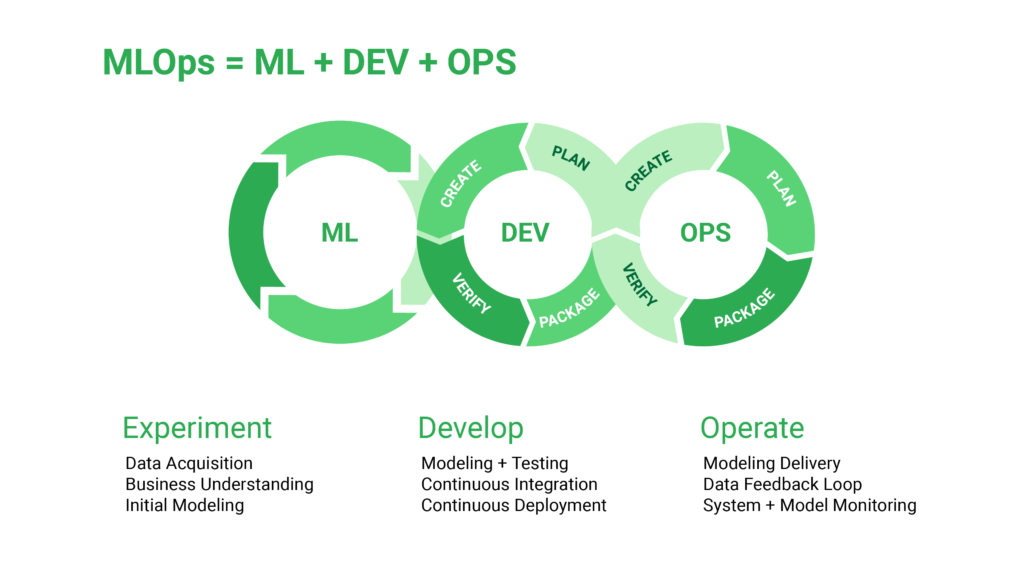
**Du développement à la production : Présentation de trois modèles phares de Machine Learning et introduction au MLOps**

****

**Introduction générale**

L’intelligence artificielle (IA) et l’apprentissage automatique (Machine Learning ou ML) sont au cœur des innovations qui transforment le secteur de la technologie. Ils continuent de révolutionner de nombreux secteurs en apportant des solutions novatrices pour l’analyse et le traitement de données massives, avec une application particulièrement adaptée dans le domaine du traitement du langage naturel ou, en anglais, natural language processing (NLP). Dans cet article, nous explorerons, dans une première partie trois techniques phares qui ont marqué ce domaine : la régression logistique, les LSTM (Long Short-Term Memory), et BERT (Bidirectional Encoder Representations from Transformers), en mettant l’accent sur leur utilisation via la bibliothèque Transformers de Hugging Face.

Ensuite dans une deuxième partie nous aborderons un concept fondamental dans le domaine de l’intelligence artificielle car il vise, à la fois, à unifier et à simplifier le processus de développement et de déploiement des modèles de Machine Learning : il s’agit du MLOps.

En effet, cette discipline émergente qui se situe à mi-chemin entre le Machine Learning et les opérations informatiques (Ops), le MLOps ne se contente pas de mettre l’accent sur la création de modèles mais englobe le cycle de vie complet : de la conception à la production et à l’optimisation continue.

Plus particulièrement, cette partie de l’article explorera les principes fondamentaux de MLOps, les étapes clés de sa mise en œuvre et comment le suivi de performance en production, via des outils comme Azure Application Insights peut consacrer l’amélioration continue des modèles.

**1)Révolutionner le traitement du langage naturel : zoom sur la régression logistique, les LSTM et BERT**

**La régression logistique : un pilier de la classification**

La régression logistique est souvent le point de départ pour ceux qui s’initient au Machine Learning. Bien qu’elle soit une méthode traditionnelle, elle reste extrêmement puissante pour la classification supervisée, qu’elle soit binaire ou multi classe. Ce modèle statistique prédit la probabilité qu’une observation appartienne à l’une des catégories fixées, cette probabilité est basée sur une fonction logistique.

**Application en NLP**

Dans le cadre du NLP, et plus spécifiquement dans l’analyse des sentiments, la régression logistique évalue si un texte véhicule un sentiment positif ou négatif. Les coefficients du modèle sont ajustés en fonction des données d’entrainement à travers la méthode de descente de gradient, optimisant ainsi la fonction de perte pour minimiser les erreurs de classification.

**LSTM : La réponses aux limites des RNN**

Les réseaux de neurones récurrents (RNN) sont un choix naturel pour le traitement des séquences de données comme le texte ou les séries temporelles. Cependant, les RNN standard ont du mal à capturer des dépendances à long terme dans les données en raison du problème de disparition du gradient. Les LSTM, une extension des RNN, pallient ce problème grâce à leur architecture sophistiquée qui permet de traiter efficacement des séquences plus longues et complexes.

**Architecture des LSTM**

Les LSTM sont dotés de trois portes régulant le flux d’informations : la porte d’entrée, la porte de l’oubli et la porte de sortie. Ces mécanismes permettent au modèle de décider quelles informations conserver et quelles informations éliminer et ce à chaque étape du calcul. Ceci rend les LSTM particulièrement adaptés pour des applications comme la traduction automatique, la génération de texte et la retranscription de conversation où la compréhension du contexte sur de longues séquences est cruciale.

**BERT : Une révolution dans le NLP**

Développé par Google, BERT a transformé la façon dont les machines comprennent le langage humain. Contrairement aux modèles précédents qui traitent le texte de manière séquentielle, c’est-à-dire de droite à gauche ou de gauche à droite, BERT analyse le texte dans les deux directions simultanément ce qui lui permet de saisir un contexte plus riche et de produire des modèles d’une précision inégalée.

**Intégration avec Hugging Face**

L’utilisation de BERT et d’autres modèles de NLP avancés est facilitée par la bibliothèque transformers de Hugging Face. Cette plateforme permet aux développeurs d’accéder facilement à une multitude de modèles pré-entrainés et de les adapter à des applications spécifiquement sans avoir à entrainer nos modèles à partir de zéro, économisant ainsi des ressources précieuses et rendent ces technologies avancées accessibles à un public plus large.

**2) La démarche orientée MLOps mise en œuvre : un guide complet**

**Principes fondamentaux du MLOps**

Le MLOps repose sur plusieurs principes clés :

1. Le MLOps c’est avant tout une culture de collaboration, toute la démarche consiste à optimiser un travail de groupe afin d’être plus efficace.
2. Automatisation : automatiser le plus possible afin de simplifier et de fiabiliser le processus dans le temps.
3. Intégration continue : les frameworks MLOps s’intègrent aux pipelines CI/CD (Continuous Integration/Continuous Deployment) ce qui permet d’automatiser le déploiement et les tests. De plus les codes étant versionnés et intégrés en continue, cela permet d’assurer la cohérence et la reproductibilité.
4. Livraison continue : la capacité de déployer de nouvelles versions de modèles en production rapidement et de manière fiable afin de palier à des problèmes comme le data drift.
5. Surveillance continue : le suivi en temps réel de la performance des modèles (monitoring) pour garantir qu’ils fonctionnent comme prévu et pour faciliter les améliorations rapides.

**Etapes clés de la mise en œuvre du MLOps**

**Tracking**

La traçabilité des expériences est fondamentale dans le MLOps. Il s’agit de documenter tout ce qui concerne le processus d’apprentissage automatique. Le tracking permet de suivre les métriques utilisées pour les modèles ce qui nous donne la possibilité de comparer les différentes versions testées et d’en garder la meilleure. Plusieurs outils existent comme MLflow, Metaflow et Kubeflow. Ils permettent, entre autres, de centraliser, suivre et visualiser ces informations.

**Stockage et gestion de versions des modèles**

Les modèles doivent être stockés de manière sécurisée en permettant à chaque instant un accès aux différentes version disponibles. Le versionnage permet de suivre les modifications au fil du temps et de revenir facilement à une version antérieure en cas de problèmes. A l’instar des outils de gestion de version de code dans le développement logiciel comme Github, il existe plusieurs solutions pour versionner les modèles de Machine Learning comme DVC (Data Version Control) qui lorsqu’il est associé à des plateformes comme Azure via Azure Blob Storage constitue un très bon outil de versionnage de modèle de ML.

**Test unitaires**

Les tests unitaires sont très importants car ils garantissent la qualité et la fiabilité des modèles de Machine Learning. Ces tests permettent de vérifier que chaque composante du code fonctionne correctement et ce de manière isolée. Cela revient à tester chaque portion du code de manière indépendante afin de détecter rapidement les erreurs et les régressions ce qui est crucial dans un environnement de production.

**Déploiement**

L’intégration de MLOps dans le processus de déploiement de modèles de ML est un processus essentiel pour transformer des modèles expérimentaux en des applications robustes, fiables et scalables en production.

En ce sens, et à l’instar du DevOps, le MLOps utilise l’intégration continue (CI) et le déploiement continue (CD) pour automatiser le déploiement une fois l’étape de tests est validée.

Concrètement, les choses se passent de la manière suivante : un pipeline qui englobe le cycle de vie de l’application ML du développement à la production et au-delà au suivi et à l’amélioration continue est créé. Ce pipeline permet d’automatiser tout le processus et ainsi chaque modification effectuée sur le code est poussée automatiquement en production une fois toutes les vérifications nécessaires validés.

**Suivi de la performance en production avec Azure Application Insights**

Une fois en production, le suivi de la performance des modèles est essentiel. En effet, cela permet de s’assurer que le modèle se comporte correctement et permet la détection d’anomalies. Azure Application Insights offre, en ce sens, des outils de monitoring très puissants. Il permet de collecter des traces, des journaux et des métriques d’utilisation ce qui permet de faciliter la détection d’anomalies par la mise en place d’alertes basées sur des critères prédéfinis.

**Analyse des données et amélioration continue**

La dernière étape du cycle MLOps est l’étape de l’analyse des données collectées en production et l’utilisation de ces enseignements pour l’amélioration continue du modèle.

Cette étape est, certes, la dernière de ce long processus mais cela ne signifie pas que ce processus s’arrête là. En effet, les insights issus de cette étape permettent d’alimenter une boucle d’amélioration constante et continue du modèle. Cette approche proactive assure que le modèle reste performant face aux changements de données et aux évolutions des exigences métier.

L’outils Azure Application Insights, permettant de collecter des métriques traditionnelles de mesure de performance mais également des traces et des logs qui permettent d’évaluer le comportement du modèle en conditions réelles, ceci offre aux équipes la possibilité de suivre et de visualiser en tant réel une panoplie de données et leur permet de détecter rapidement les anomalies et d’apprécier les opportunités d’amélioration.

Dans ce cadre, il est possible d’automatiser cette partie à travers l’intégration de workflows configurés de façon automatique pour analyser les données collectées. Ces analyses se font de manière mécanique et continue ce qui permet d’identifier les tendances, les anomalies ou les changements dans les performances du modèle au fil du temps. Ceci est en mesure de nous garantir des insights générés régulièrement et disponibles en tant réel ce qui rend les équipes plus réactives lorsqu’il s’agit de prendre la décision d’apporter des correctifs.

Ce processus est rendu plus efficace à travers la mise en place d’un feedback loop qui permet d’automatiser la réponse aux insights obtenus, le modèle est alors ajusté de manière automatique.

**Conclusion**

Le monde du l’intelligence artificielle et plus particulièrement celui du Machine Learning ne cesse d’évoluer offrant au passage des possibilités infinies d’applications.

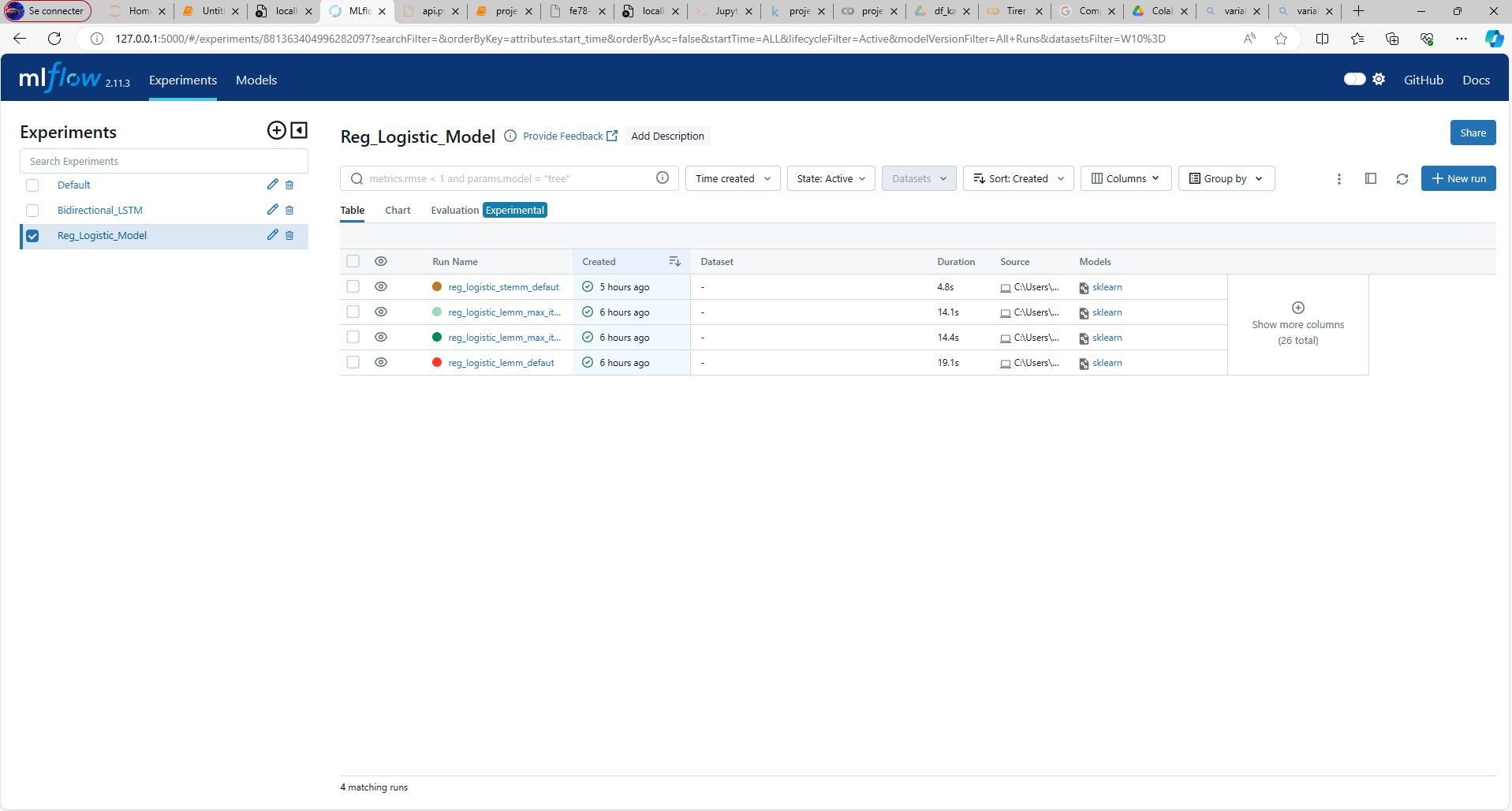
Une multitude de techniques et de modèles sont aujourd’hui plus disponibles que jamais à un large public. Des bibliothèques comme Tensorflow, Keras ou Scikit-Learn ont permis de démocratiser l’usage de modèles comme la régression logistique ou les LSTM. De même, la bibliothèque Transformers de Hugging Face a permis de faciliter l’application de BERT dans plusieurs domaines dont notamment le NLP.

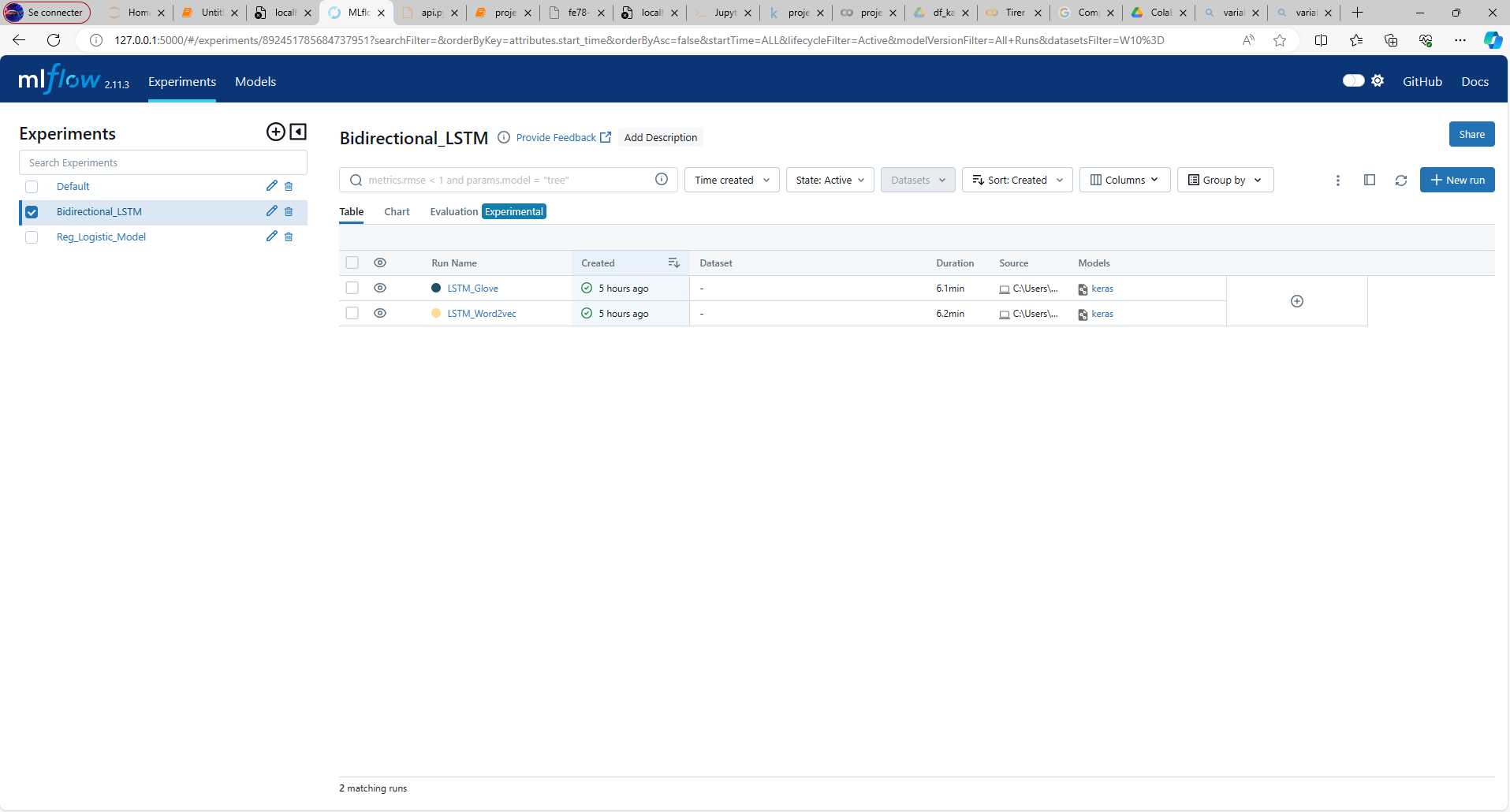
Mais du développement à la production un large fossé peut se dresser car en effet il n’est pas toujours facile de mettre en production des modèles même s’ils sont performants expérimentalement parlant. Moult projets de ML ont vu leur vie écourtée car ils échouent à l’épreuve de la mise en production.

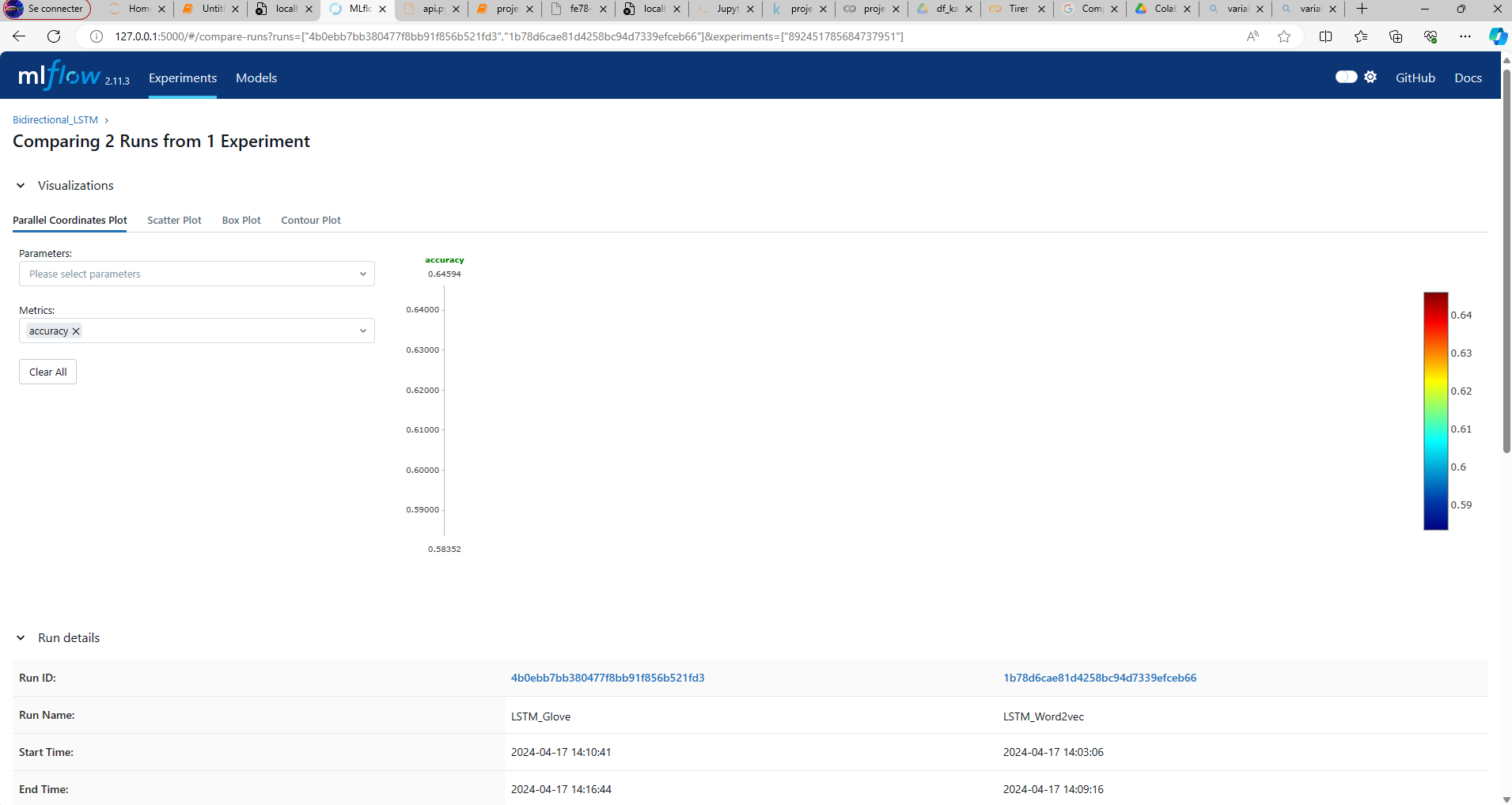
C’est dans ce cadre-là que le MLOps est apparu, à mi-chemin entre le Machine Learning et le DevOps, il permet, non seulement, d’accentuer la collaboration entre les parties prenantes du projet mais aussi une mise en production rapide et une adaptation continue aux nouvelles données et exigences. En adoptant la démarche MlOps, les entreprises peuvent, à la fois, automatiser et optimiser leurs projets ML, du développement à la mise sur le marché mais aussi garantir que leurs modèles restent pertinents et performants au fil du temps.

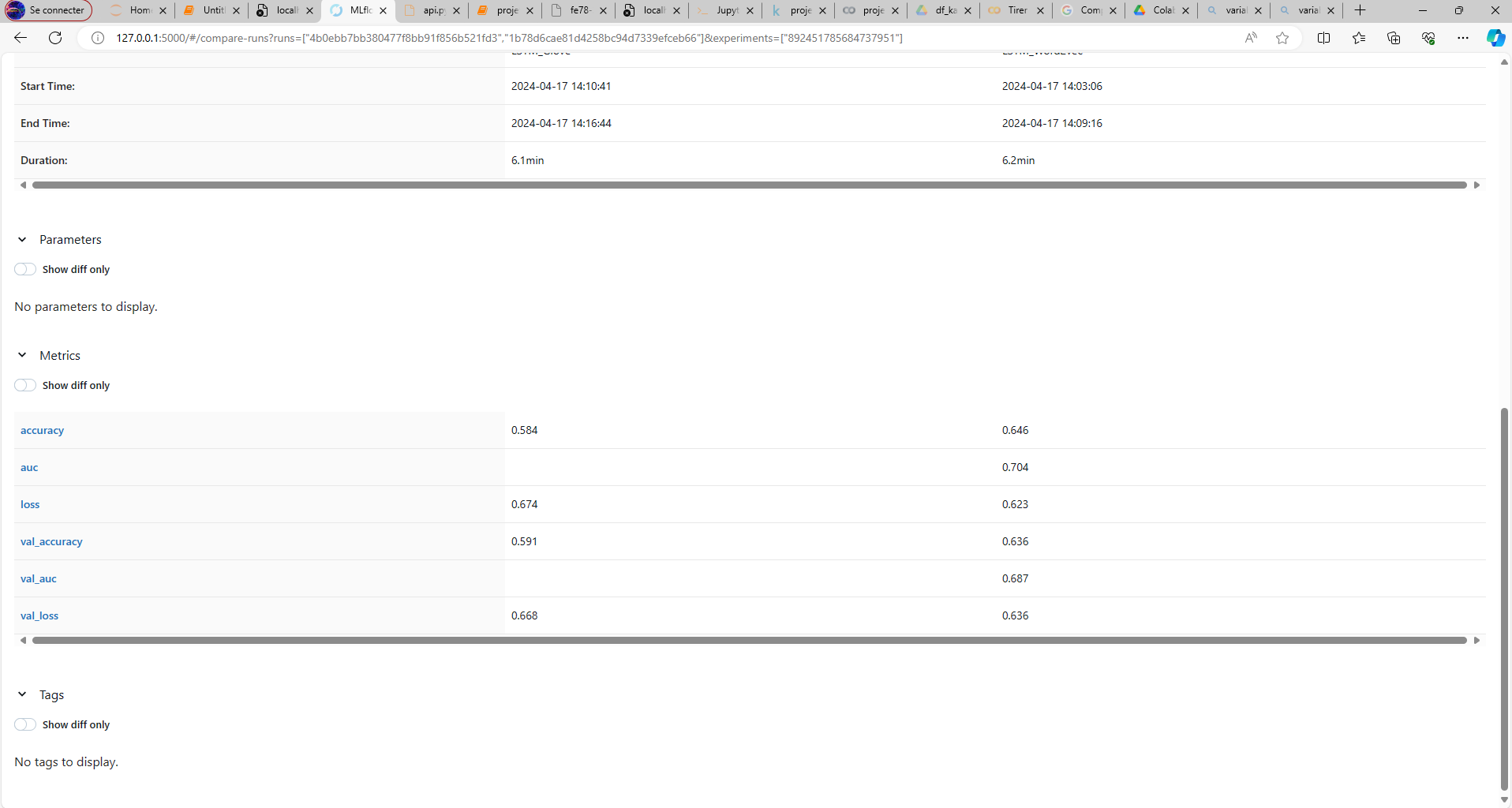
**Annexes**

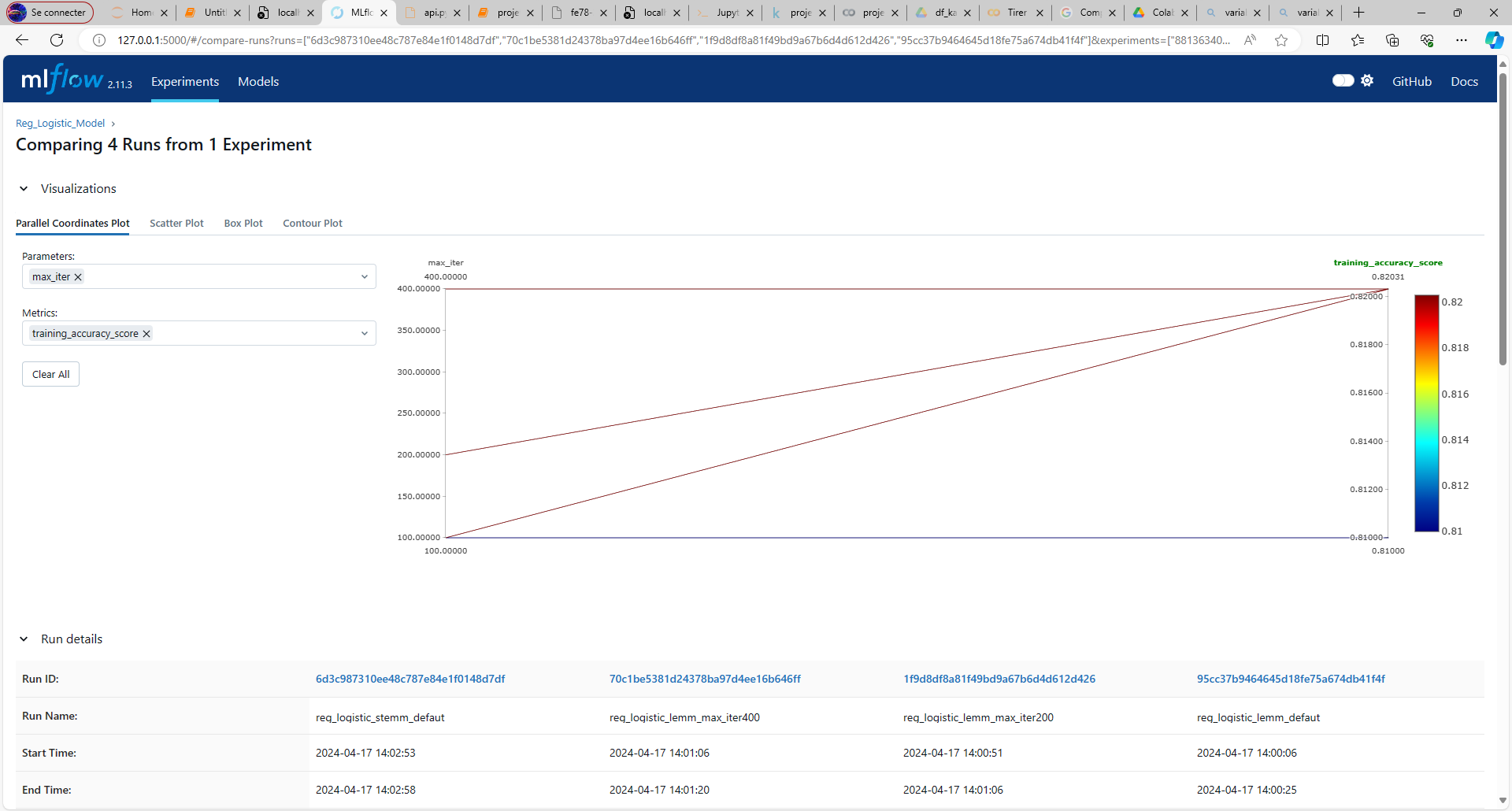
Screen shots MLflow

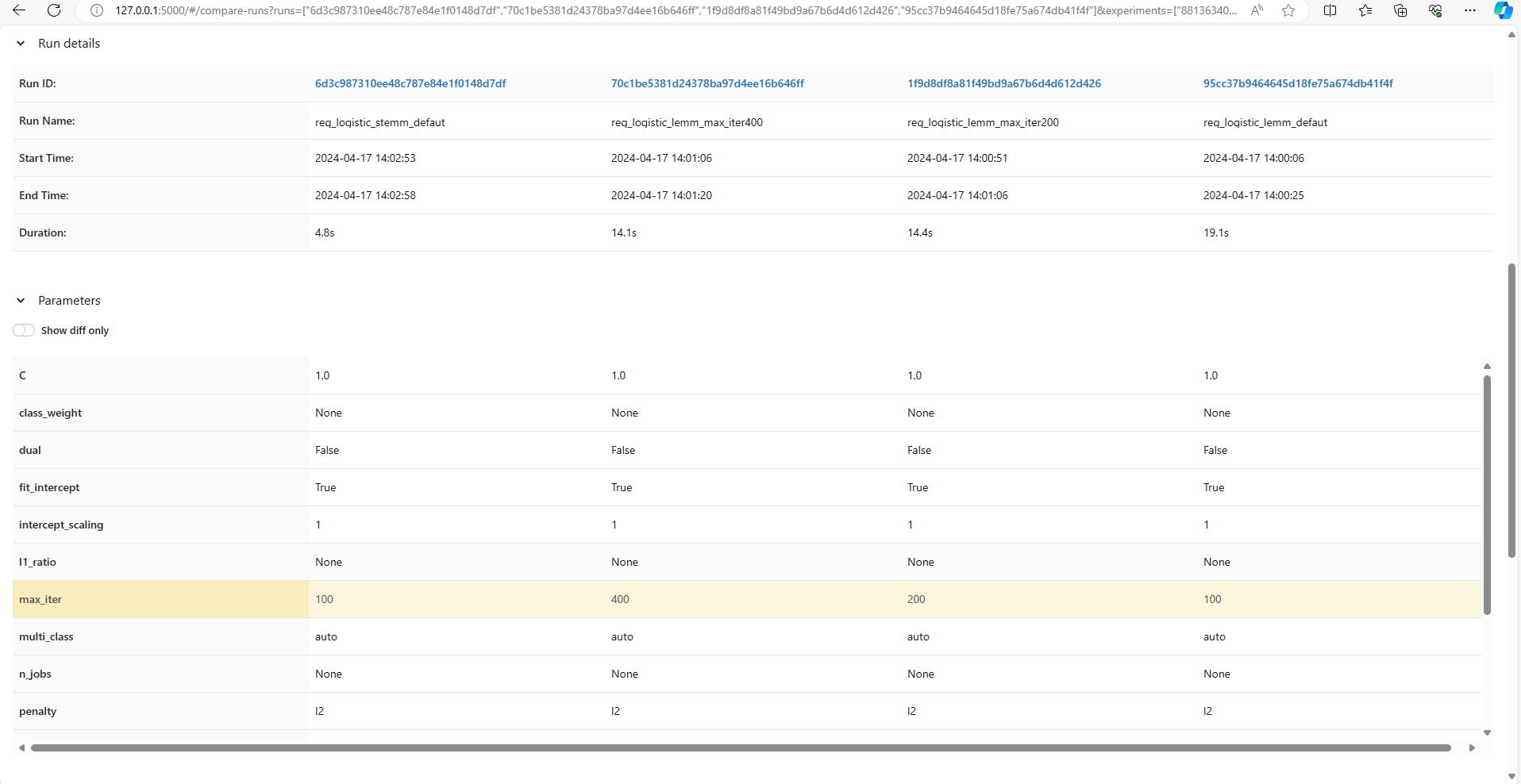


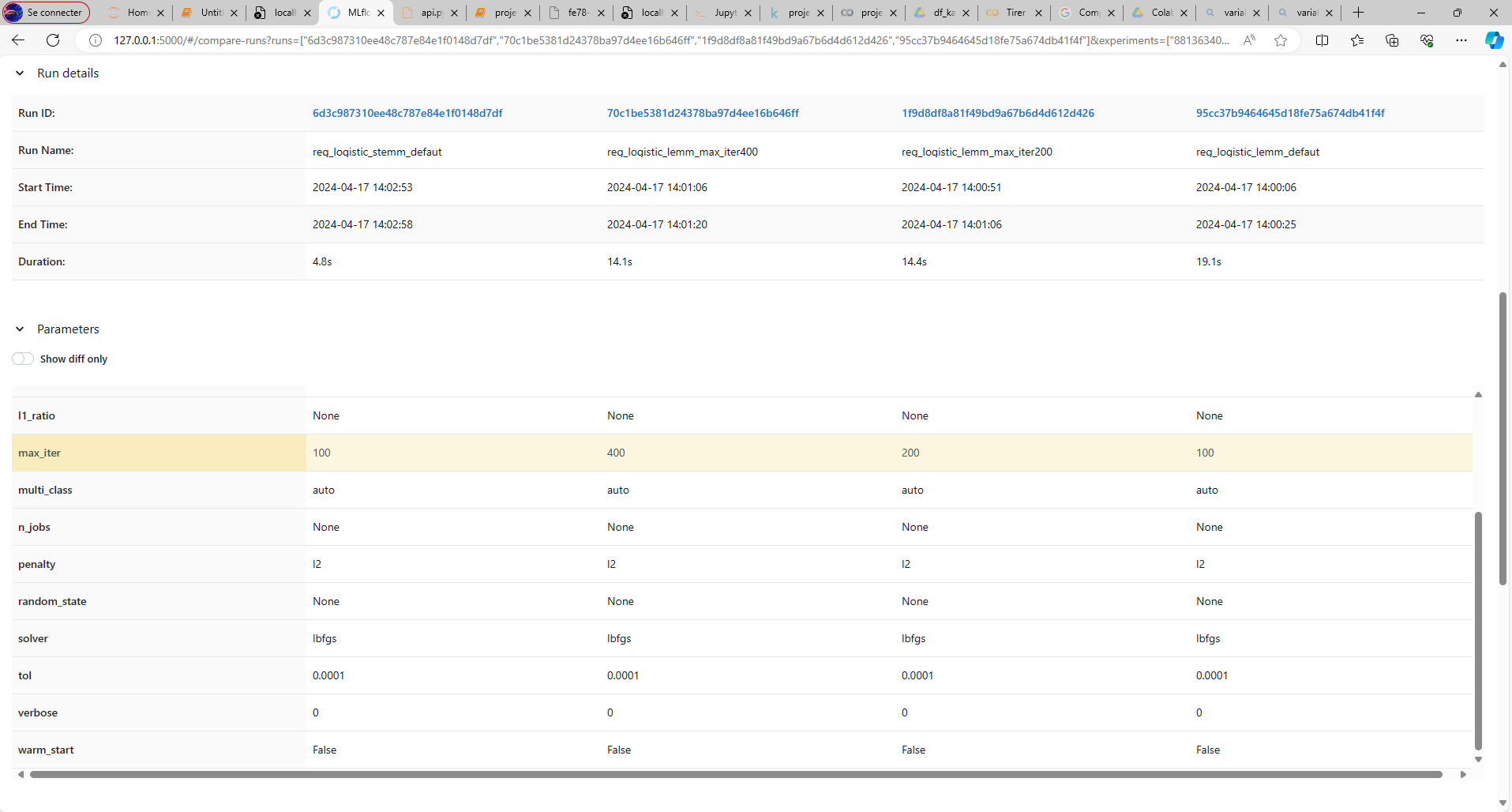


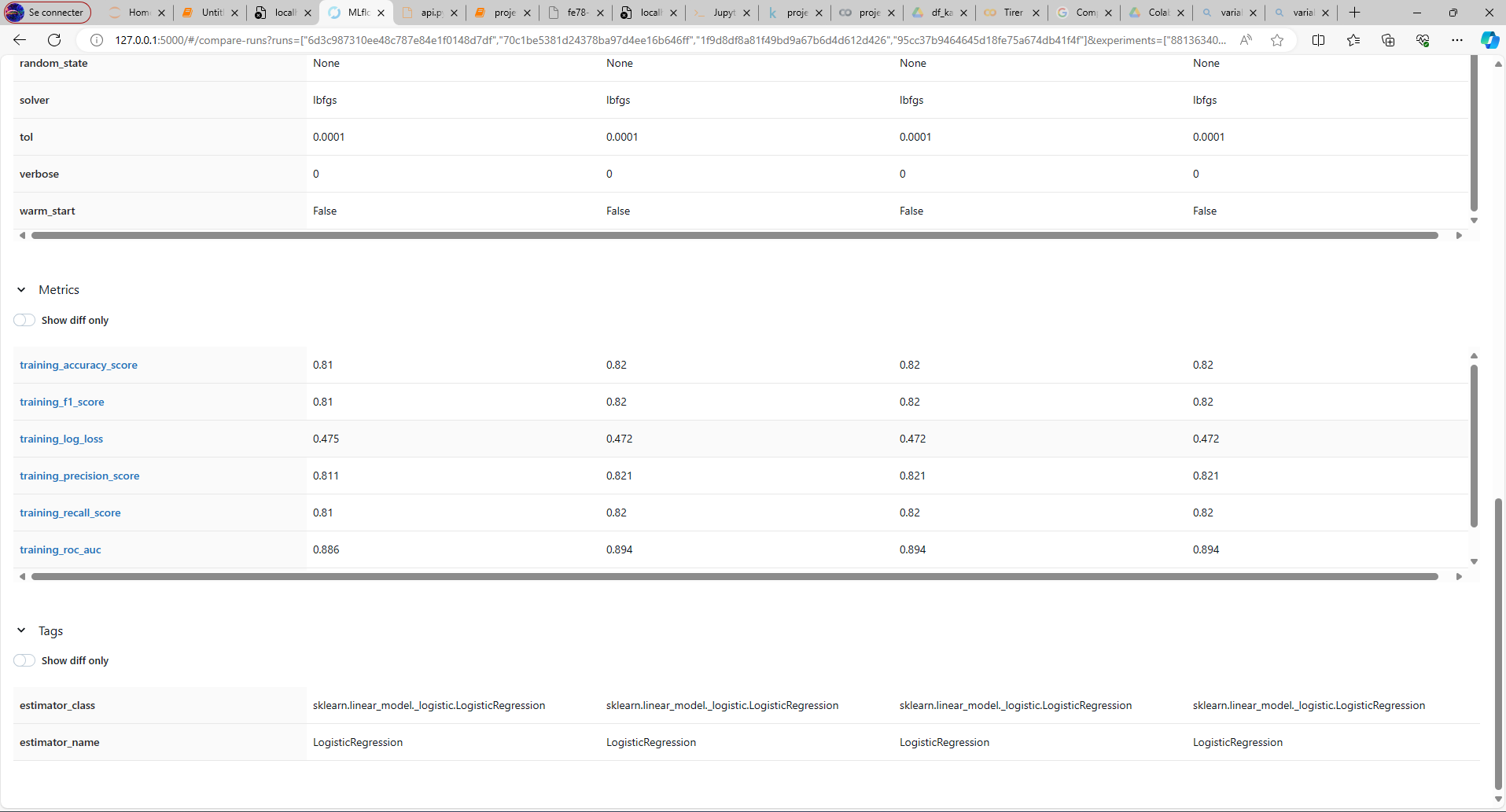


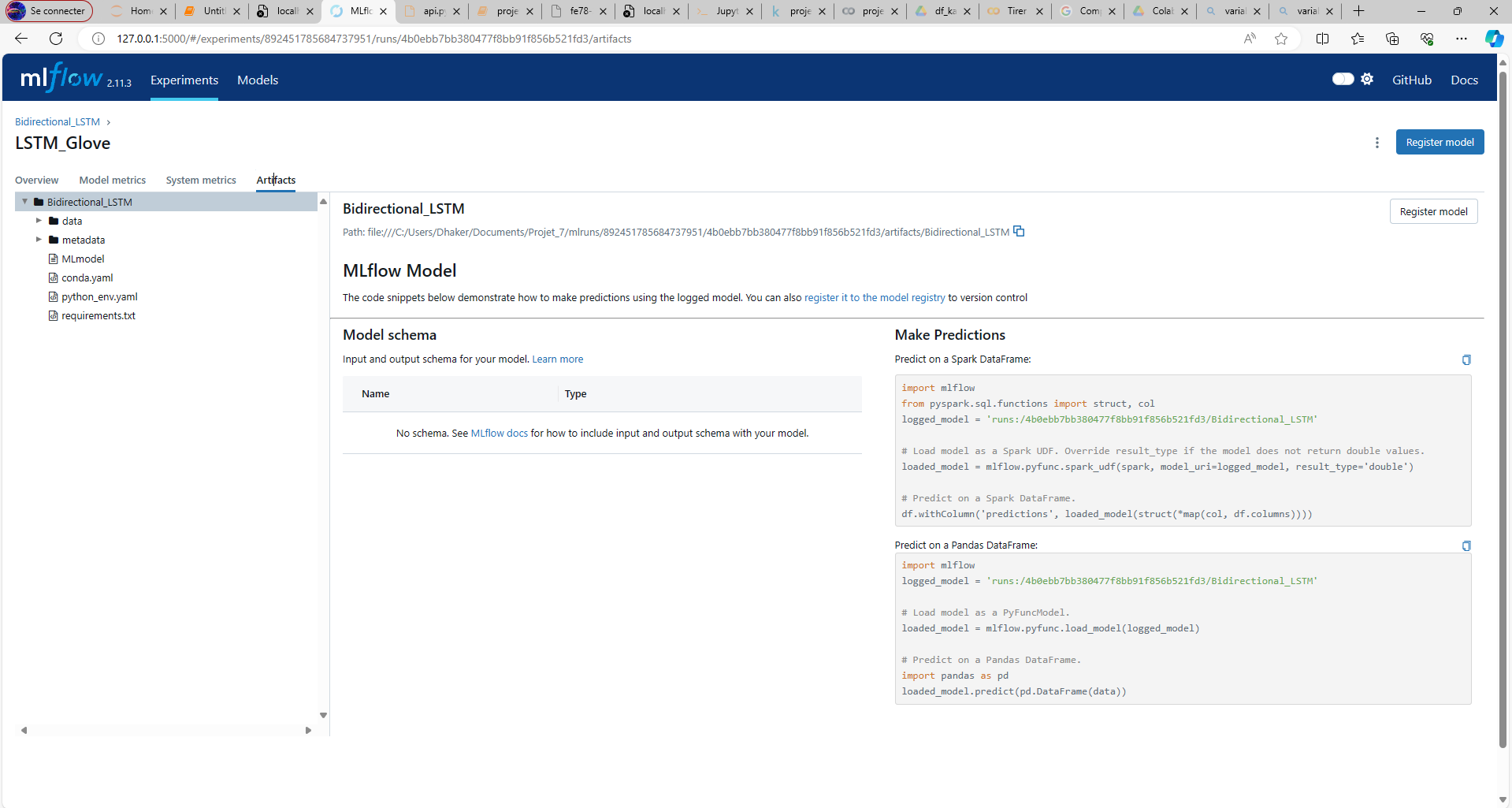


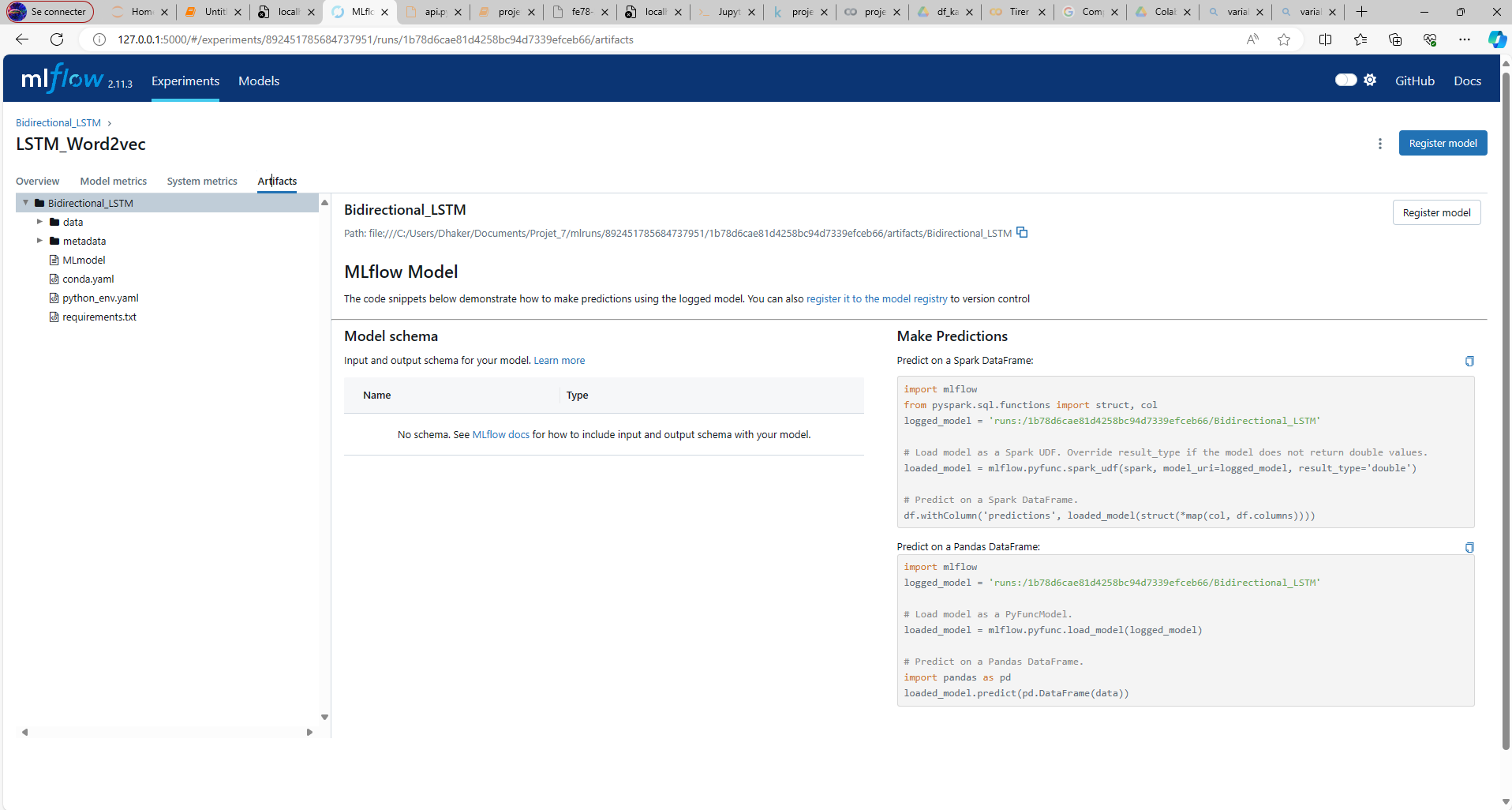


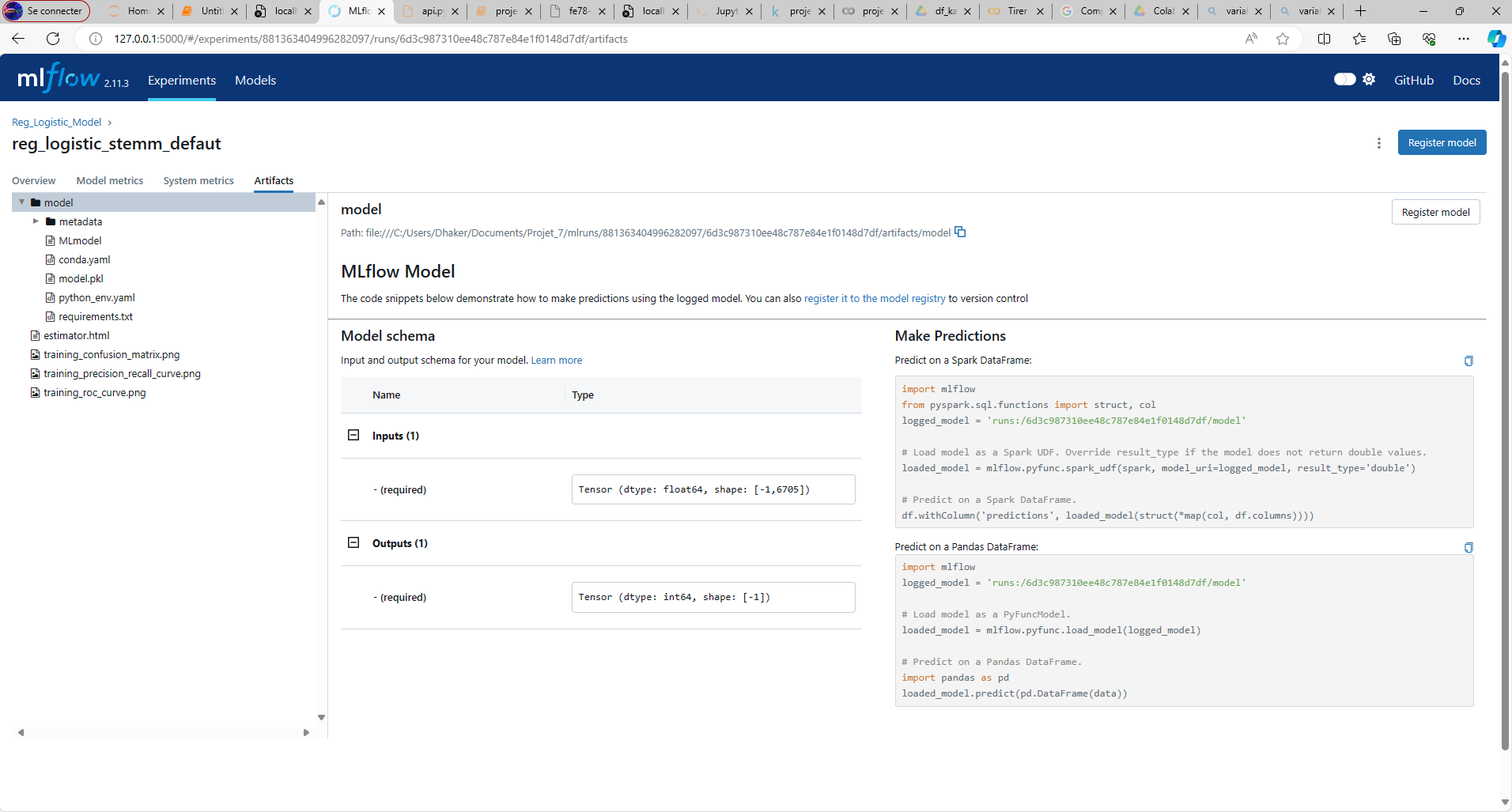


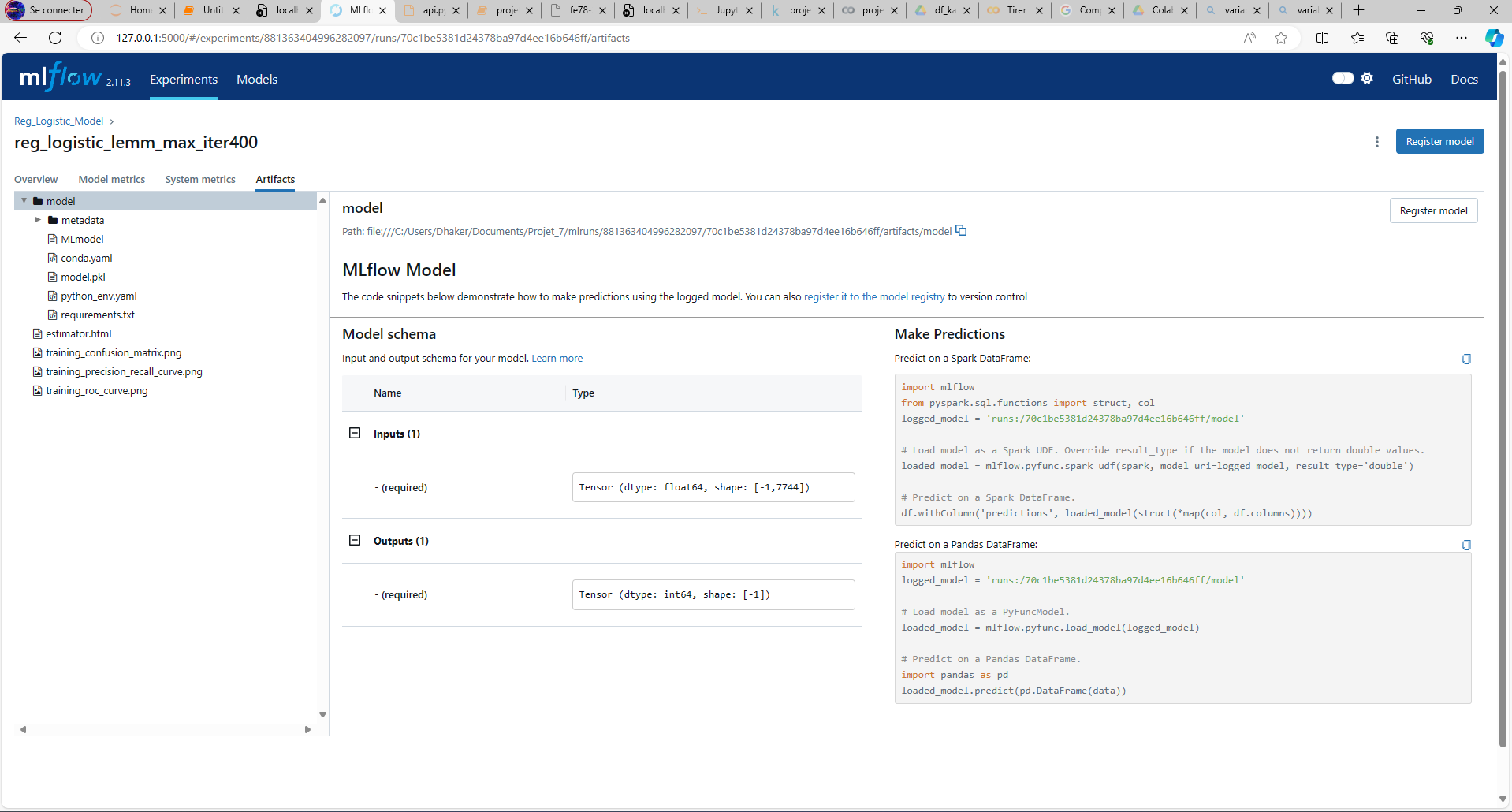


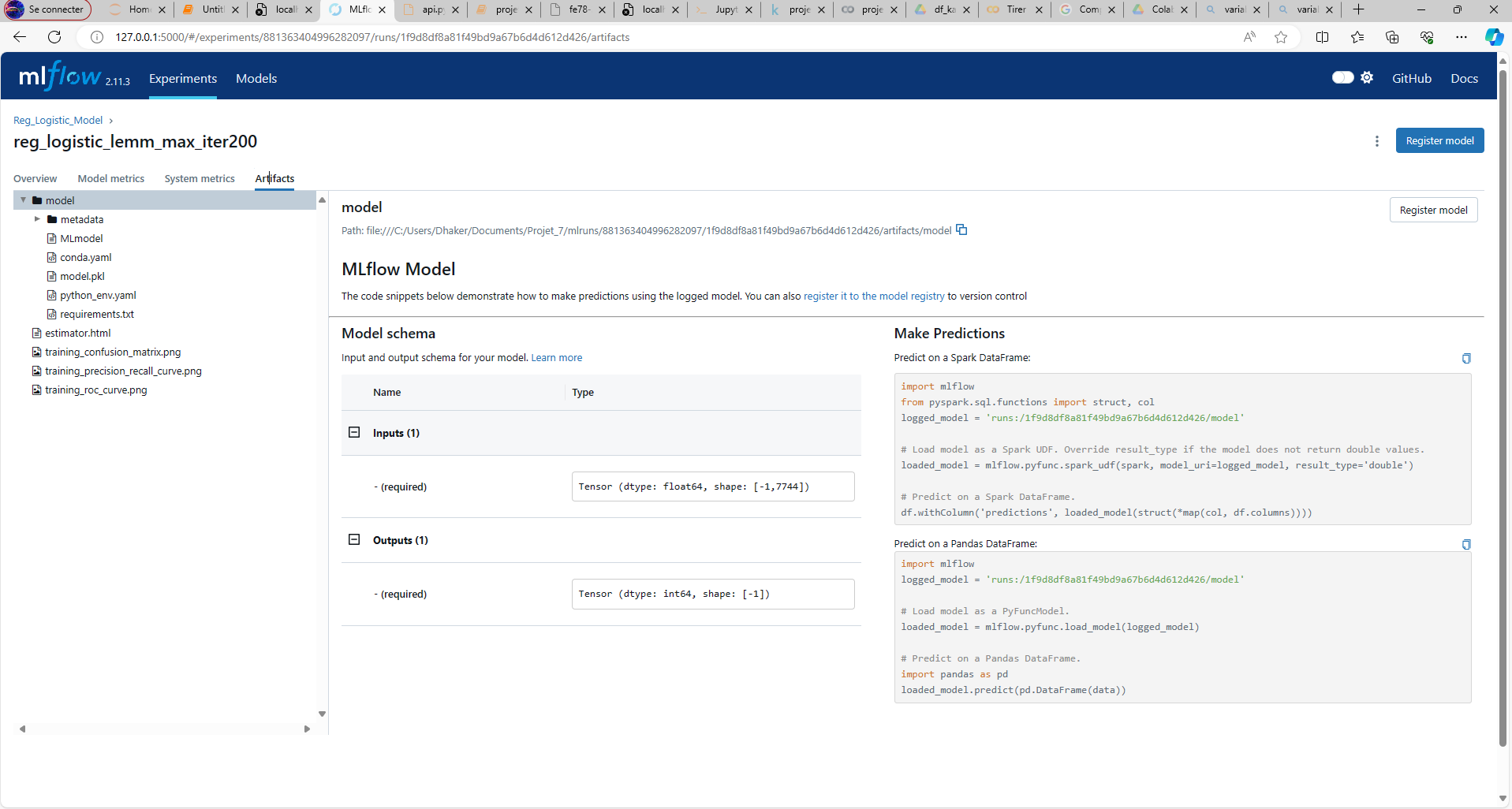


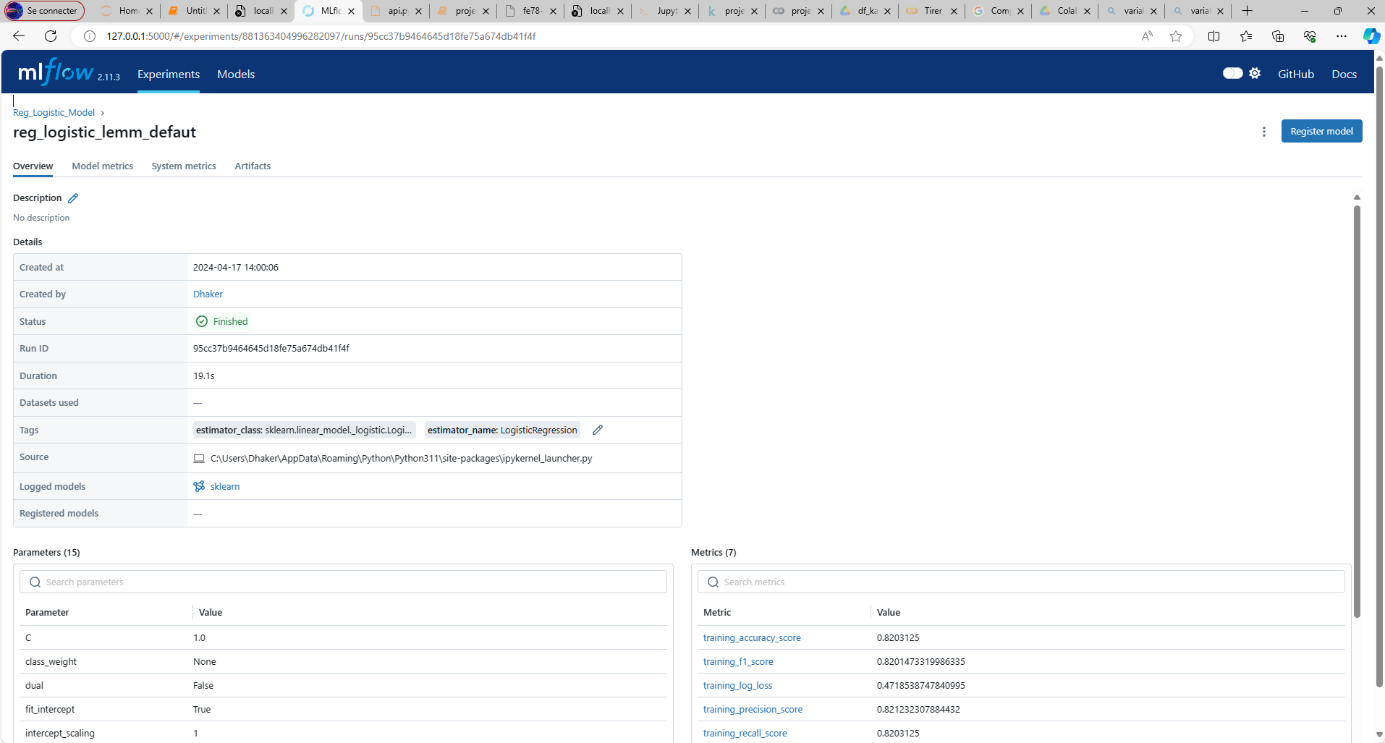


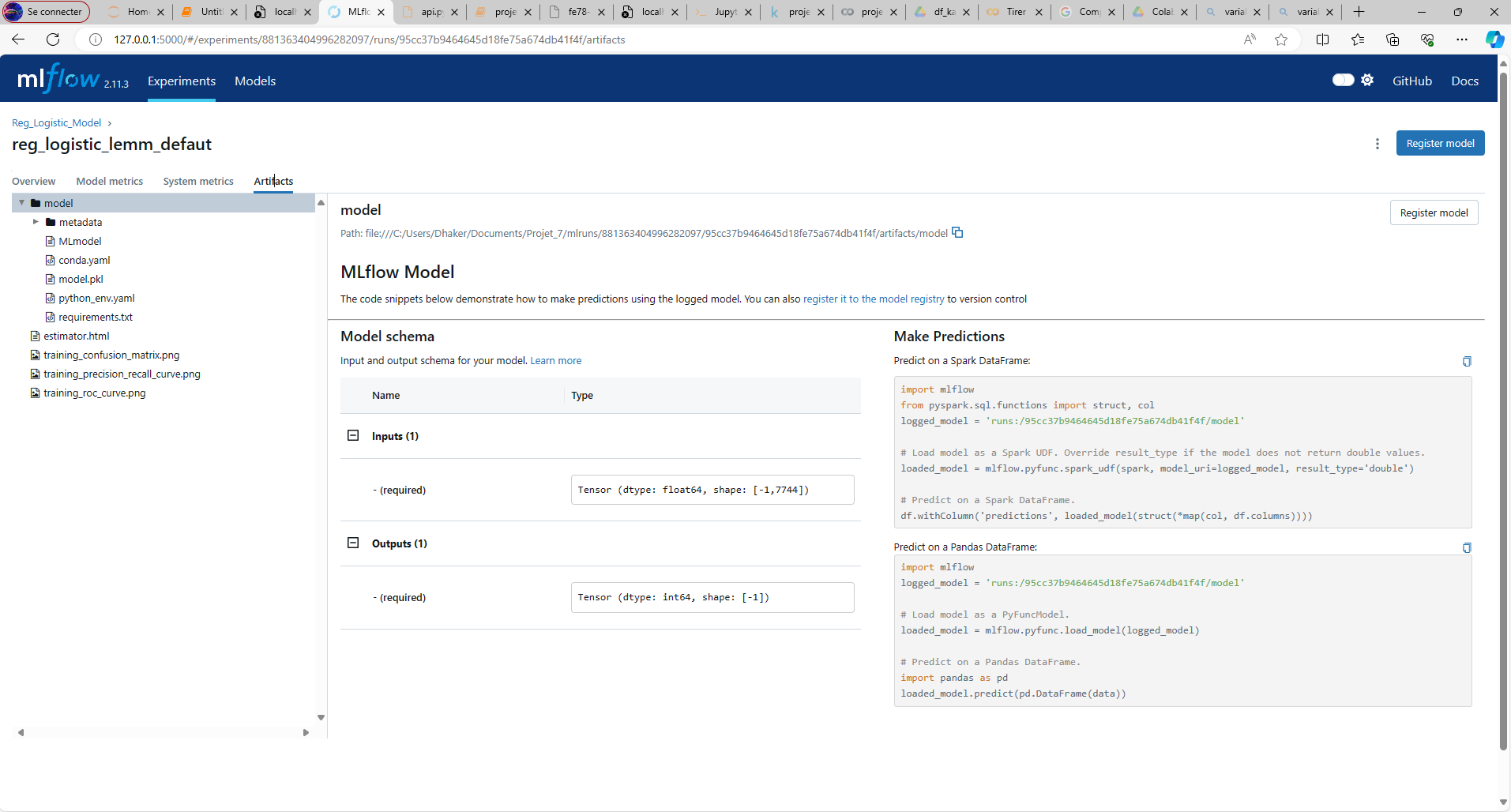




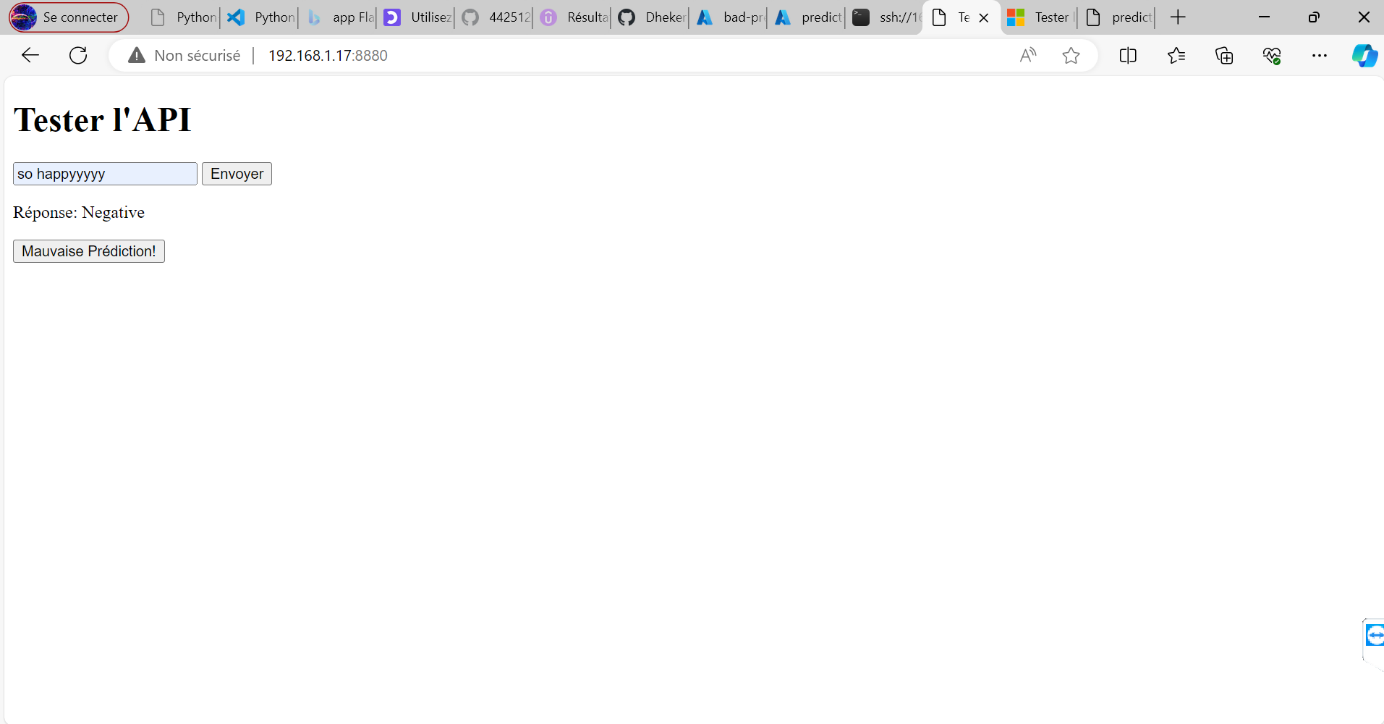






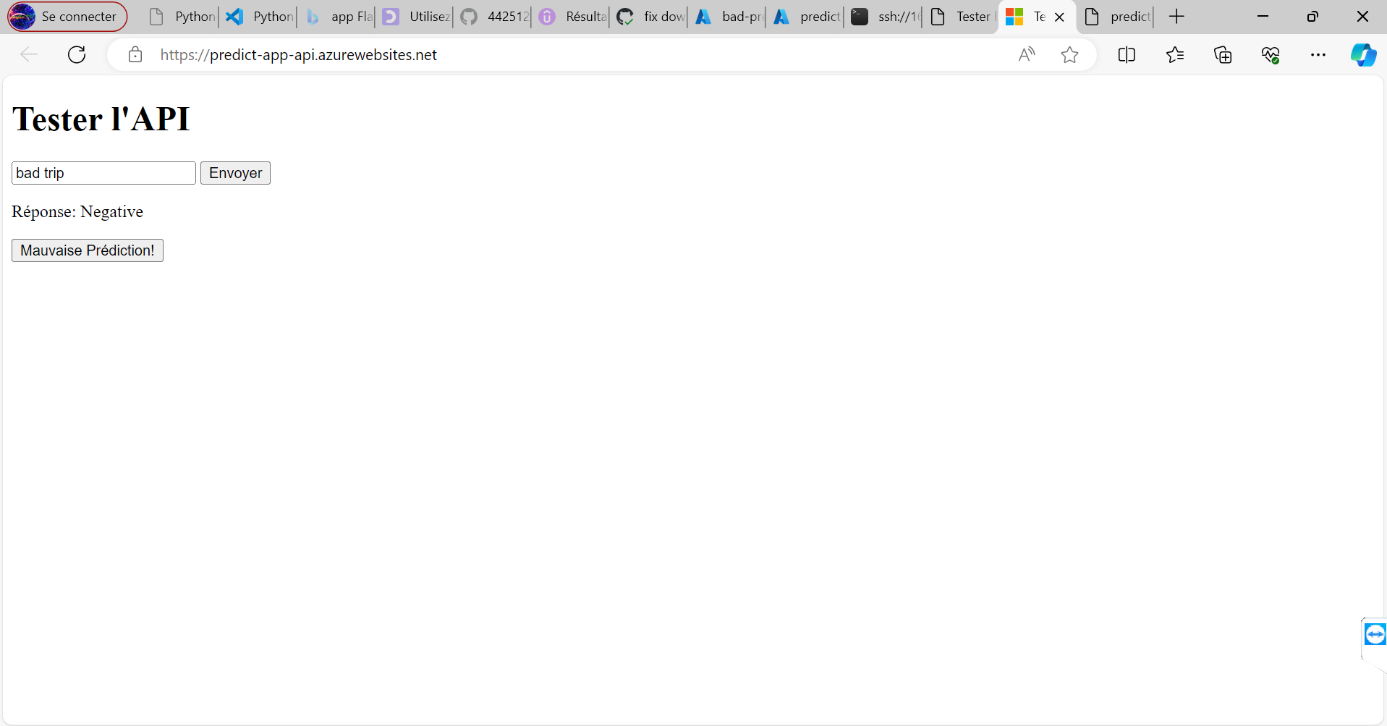


Test de l’API en local sur le port 8880.

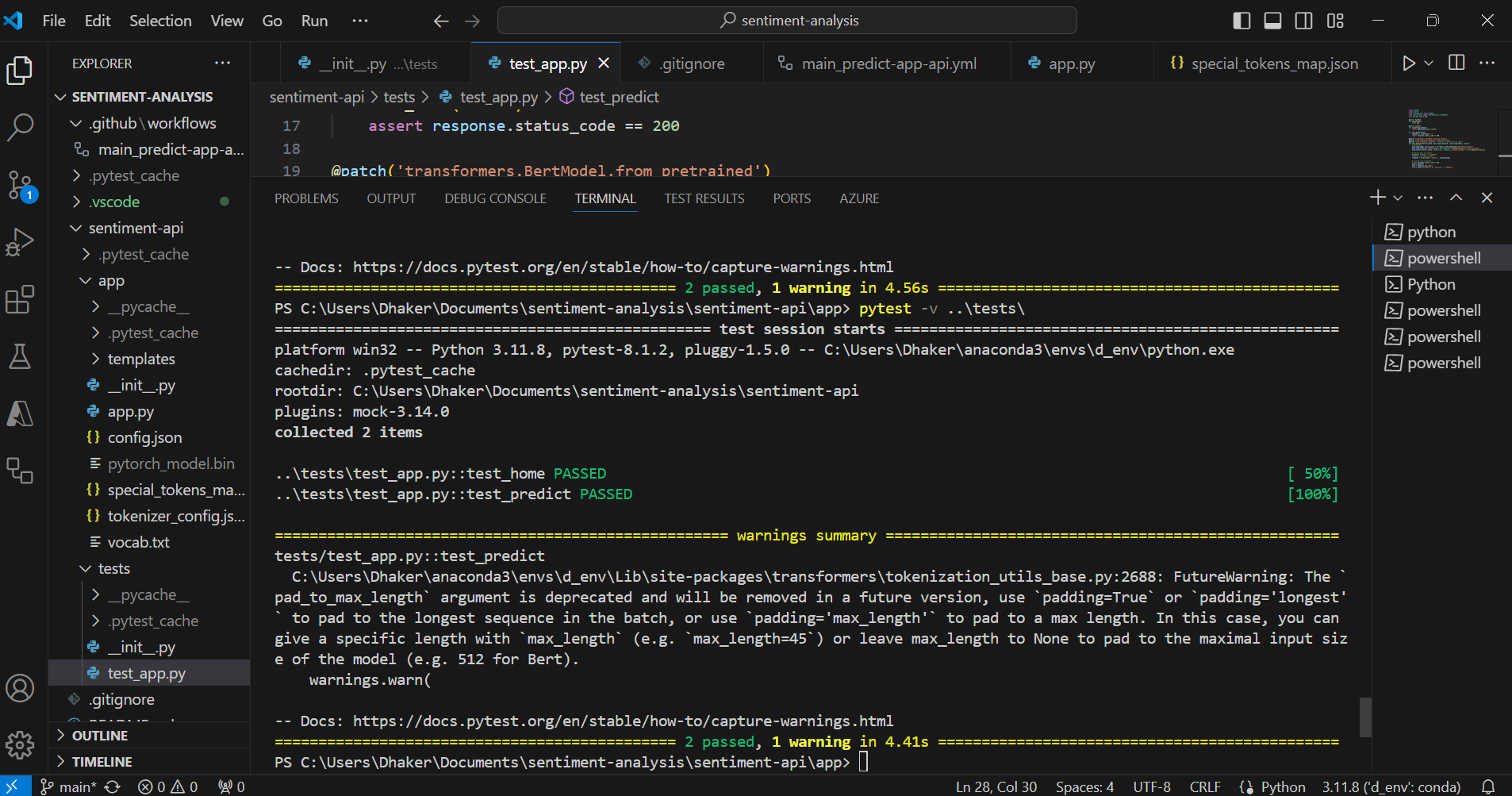


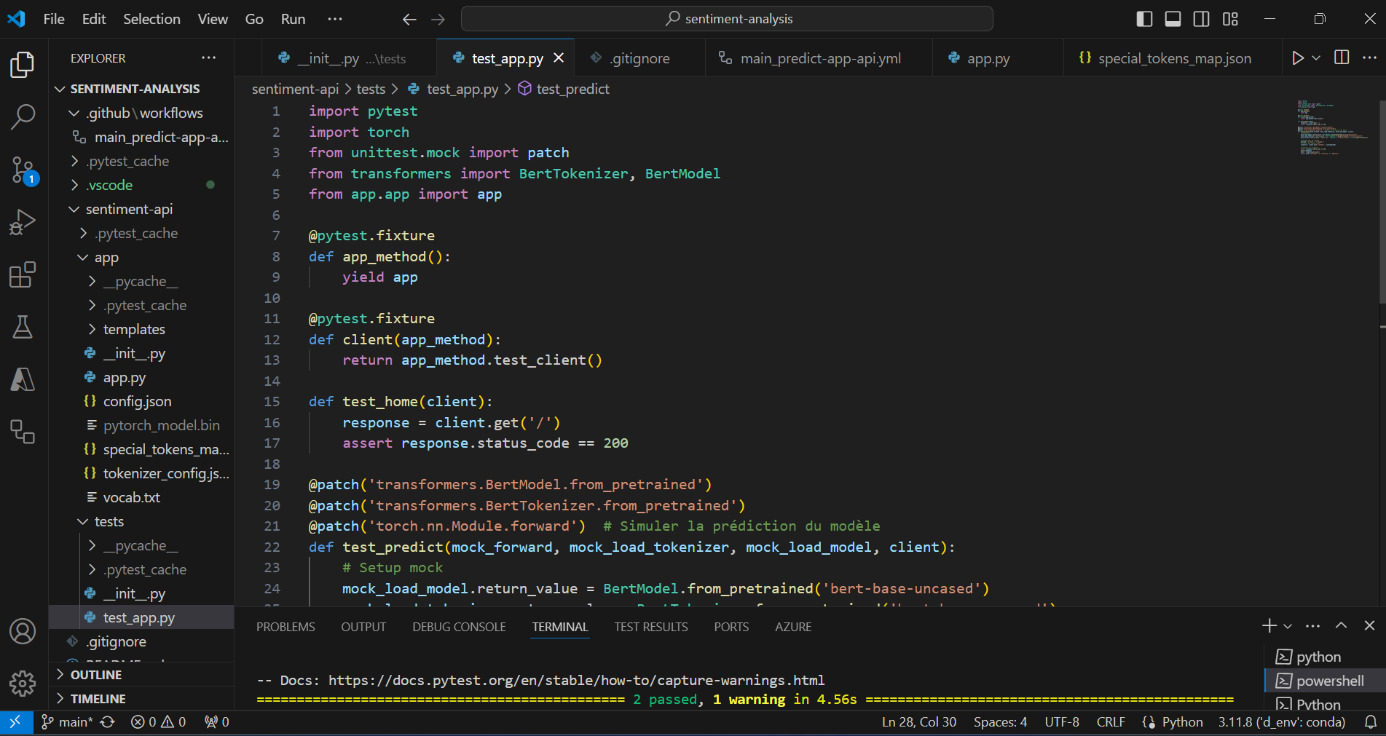
Test de l’API déployée sur Azure.





Exécution des tests unitaires en local.

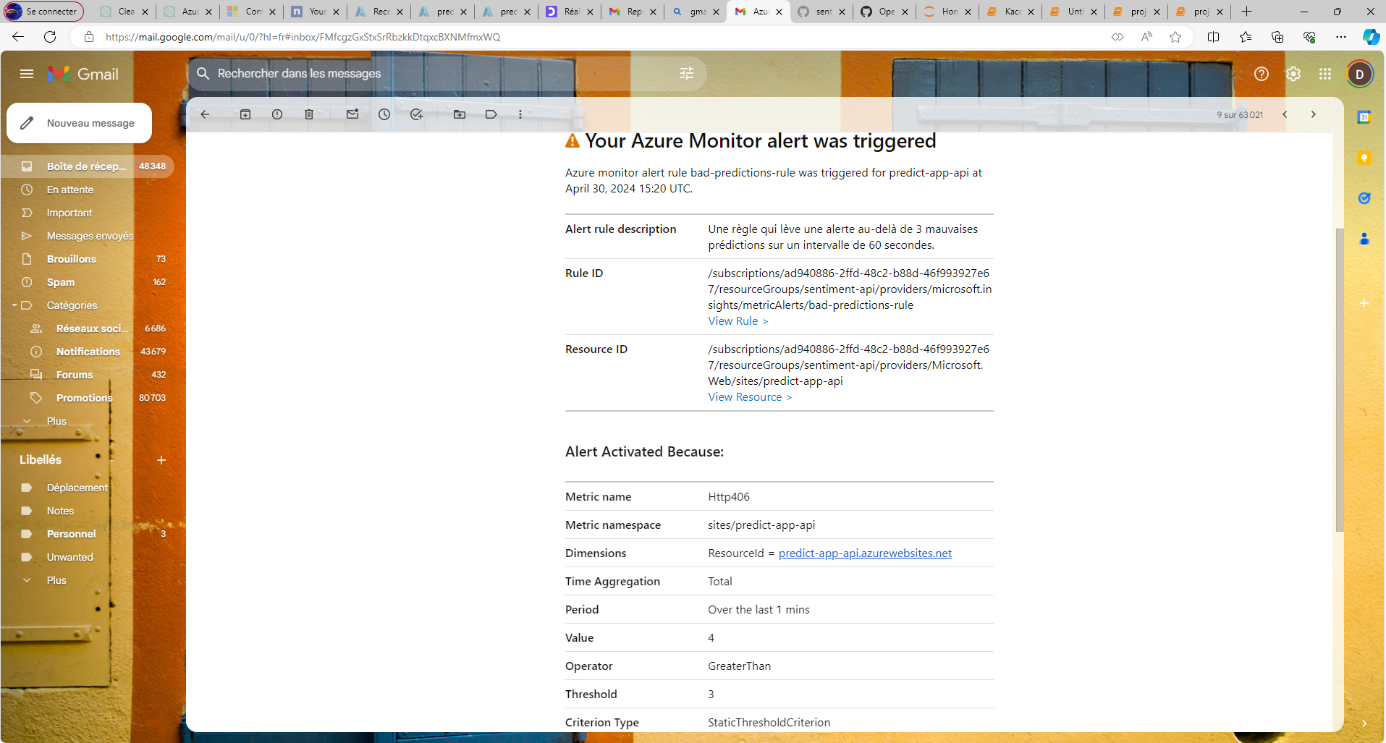




Exécution des test unitaires lors du déploiement.



Alerte reçue par mail à partir de 3 mauvaises prédiction sur un intervalle de 60 secondes.



Azure monitoring.

