

# Intelligence du Trafic Urbain : Un Projet MLOps

Présentation du projet de prédition du trafic urbain : de la donnée brute au déploiement en production, en passant par une architecture MLOps robuste. Notre objectif est de transformer les données en informations exploitables pour optimiser la mobilité urbaine.



## L'Équipe & Les Rôles

Chefs de projet Data,  
Ingénieurs ML et Responsables  
Produit.



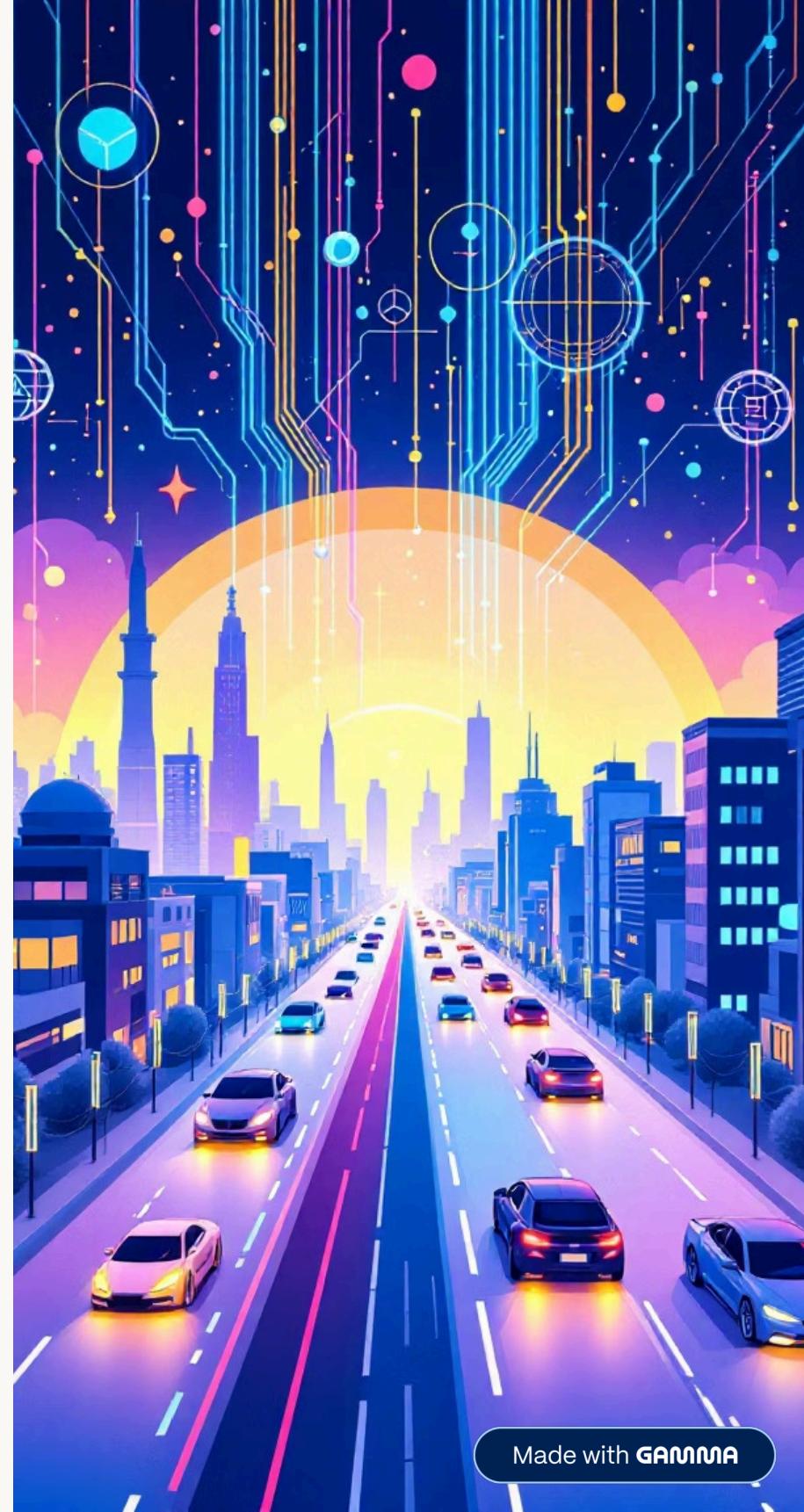
## Problématique Métier

Prédire l'affluence pour la  
gestion des flux et la  
planification des  
infrastructures.



## Objectif du Projet

Déployer un modèle fiable de prédition du trafic en temps quasi-réel.



# Collecte et Préparation des Données de Trafic

Le socle de toute prédiction réussie réside dans la qualité des données. Nous avons agrégé et traité plusieurs jeux de données publics pour construire une base solide pour notre modèle.

1

## Sources de Données

Quatre sources Open Data (capteurs de débit, axes de circulation, horodatage) fournissant des volumes importants d'informations sur la circulation urbaine.

2

## Variables Clés

Débit (nombre de véhicules), emplacement du capteur, axe routier, jour de la semaine et heure (granularité fine).

3

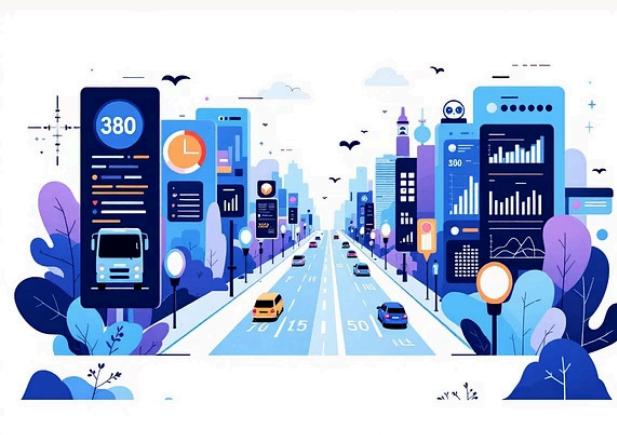
## Nettoyage & Filtrage

TraITEMENT DES VALEURS MANQUANTES, DÉTECTION DES ANOMALIES DE CAPTEURS ET STANDARDISATION DES FORMATS POUR L'ENTRAÎNEMENT.

4

## Risques Identifiés

Qualité des données hétérogènes et risque de biais liés aux capteurs fixes (ne couvrant pas tout le réseau).



# Conception et Entraînement du Modèle de Prédiction

Nous avons mis en place une approche comparative, démarrant par une base de référence simple pour ensuite explorer des modèles plus complexes, optimisés par des *features* pertinentes.

## Choix des Modèles

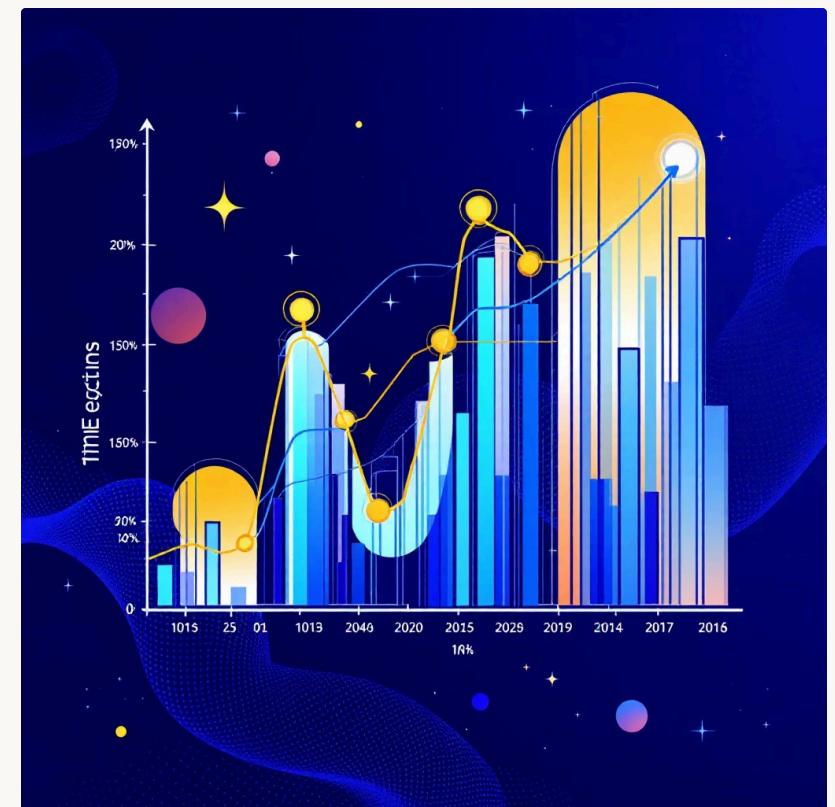
Baseline (Moyenne mobile) pour comparaison, Régression Linéaire et Modèle basé sur les Séries Temporelles (ex: LSTM ou Prophet) pour la prédiction finale.

## Ingénierie des Caractéristiques (Features)

Trafic historique (décalé), indicateurs temporels (heure, jour, saisonnalité), et facteurs exogènes (événements spéciaux).

## Méthodologie

Validation croisée temporelle et entraînement itératif. L'objectif est la généralisation sur des périodes futures non observées.

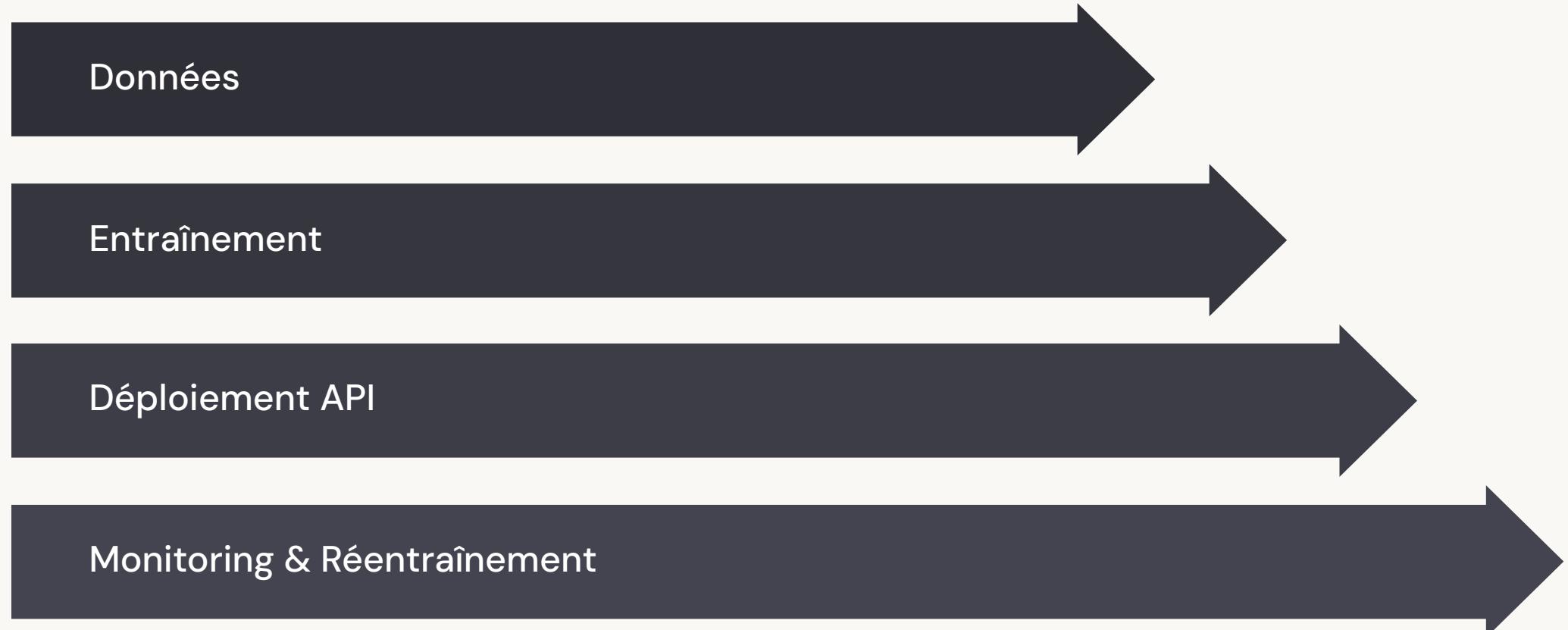


## Métrique de Performance

Nous utilisons la **MAE (Erreur Absolue Moyenne)** comme KPI principal pour mesurer l'écart moyen entre la prédiction et le trafic réel. Le **R<sup>2</sup>** est aussi surveillé.

# Architecture MLOps : Du Notebook à la Production

Pour garantir la fiabilité et la reproductibilité, notre projet s'appuie sur une architecture MLOps complète, intégrant l'entraînement, le suivi et le déploiement continu.



L'environnement Cloud (GCP / AWS Free Tier) permet un POC (Proof of Concept) à faible coût. L'automatisation est gérée par des outils d'intégration et de livraison continues.



## Cloud & Infrastructure

Infrastructure scalable sur plateforme cloud, gérant les ressources de calcul et le stockage.



## Code et CI/CD

Git pour le versionnement, Docker pour la conteneurisation, et GitHub Actions pour l'automatisation du pipeline.



## Suivi des Expérimentations

MLflow assure le tracking des hyperparamètres et des métriques, ainsi que le *Model Registry*.

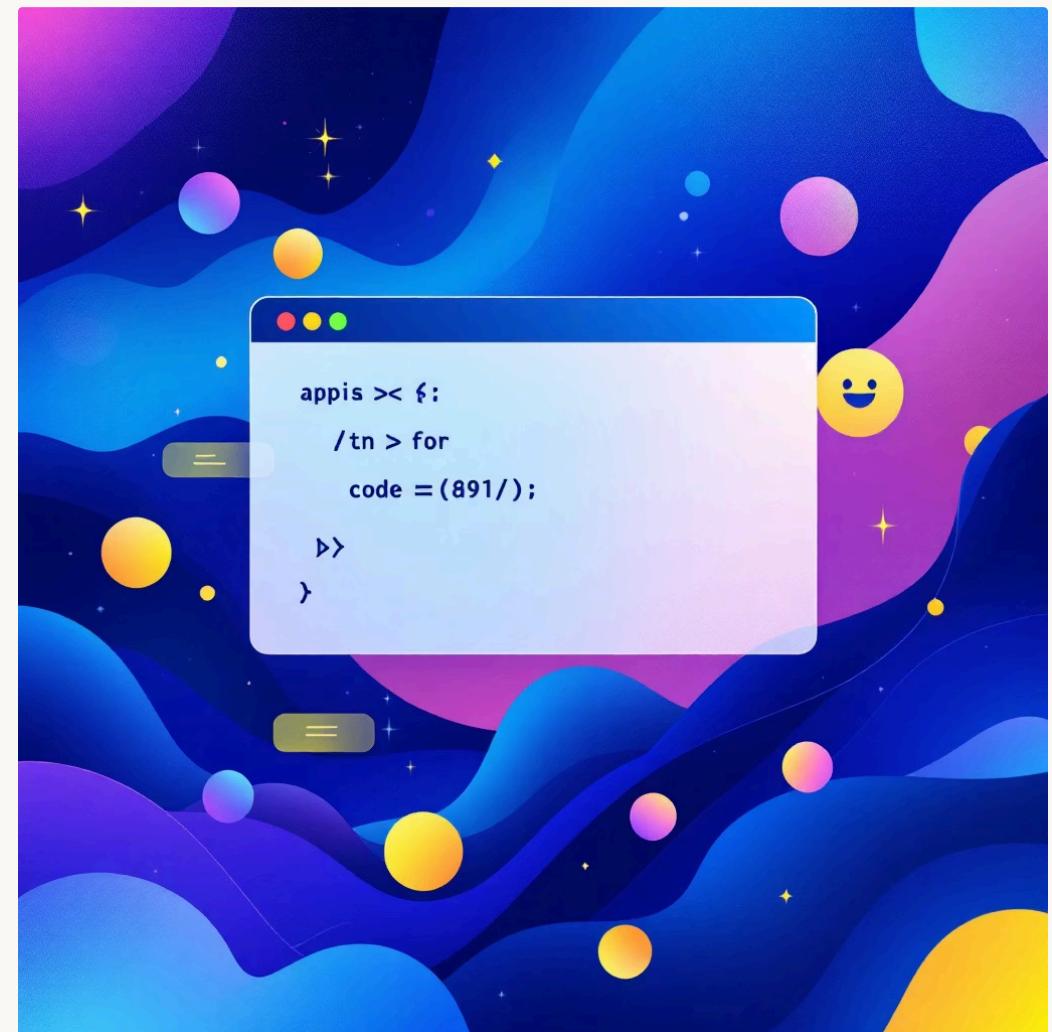
# Déploiement et Exposition via API

Le modèle entraîné est exposé comme un service web accessible via une API REST. Cette étape est cruciale pour l'intégration avec les systèmes clients.

## L'API de Prédiction (FastAPI)

- Framework **FastAPI** pour sa rapidité et sa documentation automatique (Swagger UI).
- Conteneurisation avec **Docker** pour garantir un environnement d'exécution isolé et portable.
- Endpoint principal : `/predict` acceptant les données d'entrée (capteur, heure, jour) et retournant la prédiction de débit.

- Un appel de test réussi via cURL ou Postman confirme que l'API est opérationnelle et prête à être consommée par les applications métier.





# Monitoring et Détection de Dérive (Drift)

Une fois en production, le modèle doit être surveillé en continu. Nous suivons les métriques de performance technique (latence, erreurs) et de performance métier (précision des prédictions).

## Métriques API

- 1 Latence des requêtes, taux d'erreur 5xx, et fréquence d'utilisation de l'endpoint /predict.

## Détection de Dérive

- 2 Surveillance des distributions des données d'entrée et des prédictions. Alerte en cas de trafic anormal (météo extrême, vacances).

## Dashboard (Grafana)

- 3 Visualisation en temps réel des performances et de l'état de santé de l'API et du modèle.

## Stratégie de Retrain

- 4 Déclenchement automatique d'un nouvel entraînement si la précision du modèle passe sous un seuil critique (ex. : MAE > X).

# Sécurité et Conformité Réglementaire

Bien que n'utilisant pas de données personnelles, la sécurité de l'infrastructure et la conformité sont des piliers fondamentaux de notre déploiement.



## Conformité RGPD

Confirmation qu'aucune donnée à caractère personnel n'est utilisée. Néanmoins, un registre de traitements simulé est maintenu par bonne pratique.



## Chiffrement

Communication chiffrée (HTTPS) entre l'API et les clients pour garantir l'intégrité et la confidentialité des échanges.



## Gestion des Accès (IAM)

Politiques IAM strictes pour l'accès à l'infrastructure cloud et aux secrets du déploiement (clés API).

# Optimisation des Coûts et Scalabilité

Pour garantir la viabilité économique à long terme, nous avons évalué le profil de charge estimé et les options de mise à l'échelle.

1.2s

Latence Actuelle

Mesurée sous charge standard (temps de réponse moyen).

10x

Facteur d'Échelle

Capacité à gérer 10 fois le volume de requêtes actuel via l'auto-scaling.

80%

Réduction des Coûts

Obtenue par l'optimisation des ressources cloud (utilisation du *serverless*).

**Tests de charge:** Simulation de pics de trafic pour identifier les goulets d'étranglement. L'auto-scaling permet d'ajuster automatiquement les ressources de l'API en fonction de la demande, optimisant ainsi le coût vs la performance.



# Documentation et Bonnes Pratiques MLOps

Une documentation complète est essentielle pour la maintenance, la collaboration et la passation du projet à d'autres équipes.

## Model Card & Runbook

Description détaillée des performances, des limites du modèle, et des procédures opérationnelles en cas d'incident.

## Guide de Déploiement

Instructions claires pour la réplication et le déploiement de l'infrastructure dans un nouvel environnement.

## Workflow CI/CD

Explication étape par étape du pipeline d'intégration continue et de livraison, assurant des mises à jour fiables.

## Checklist de Promotion

Procédure rigoureuse pour passer du modèle de *staging* à la *production*, incluant les vérifications de monitoring et de sécurité.

# Bilan, Améliorations Futures et Q/R

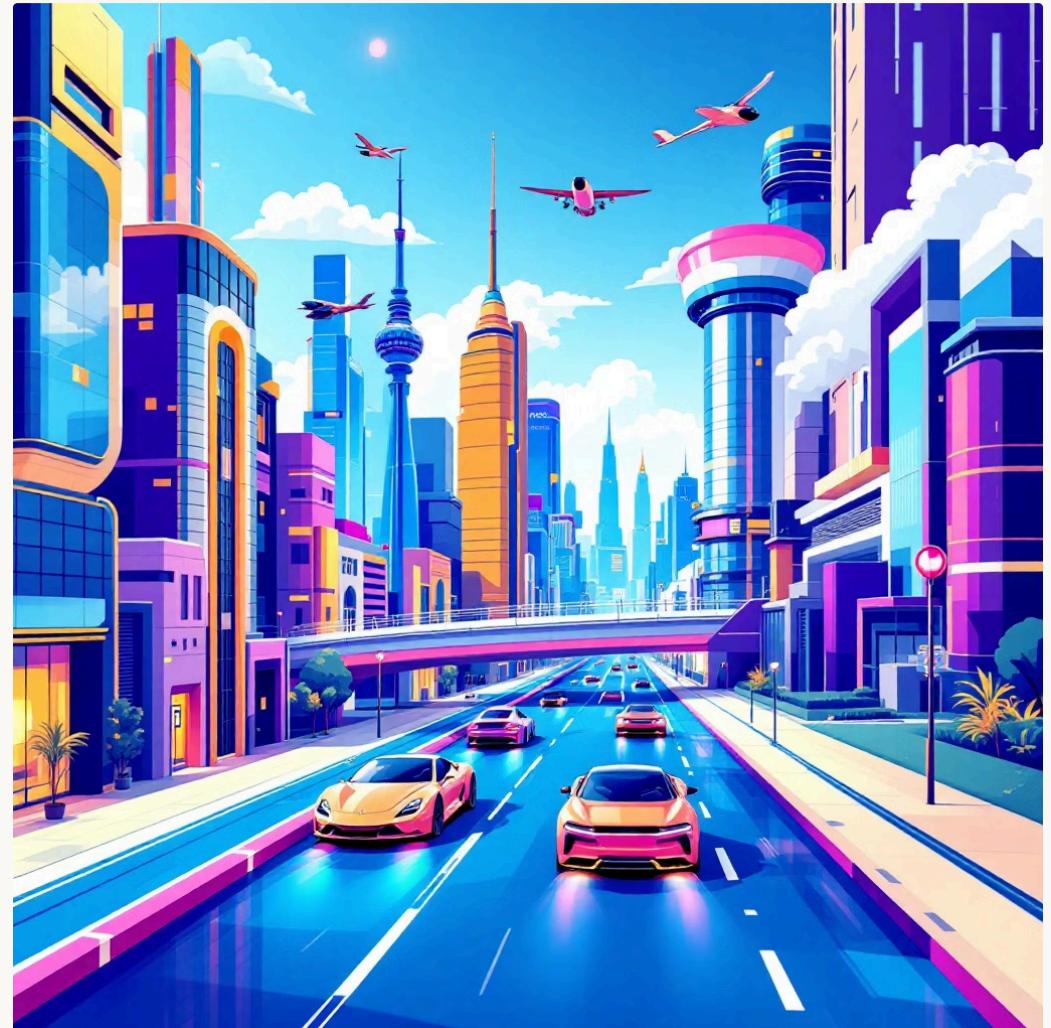
Le projet a atteint son objectif de déploiement d'un modèle de prédition du trafic en production via une architecture MLOps robuste. Cependant, l'amélioration continue est essentielle.

## Bilan et Difficultés

- Succès du déploiement conteneurisé et du pipeline CI/CD.
- Difficulté principale: Gestion de l'hétérogénéité des données de capteurs (calibration).
- MLflow a permis une gestion efficace des 45+ expérimentations d'entraînement.

## Améliorations Possibles

- Intégrer des données météorologiques et d'événements en direct (exogènes).
- Explorer des modèles plus avancés (ex: GNN pour les dépendances spatiales du réseau routier).
- Mise en place d'un système de *Human-in-the-Loop* pour valider les alertes de dérive.



## Questions & Réponses