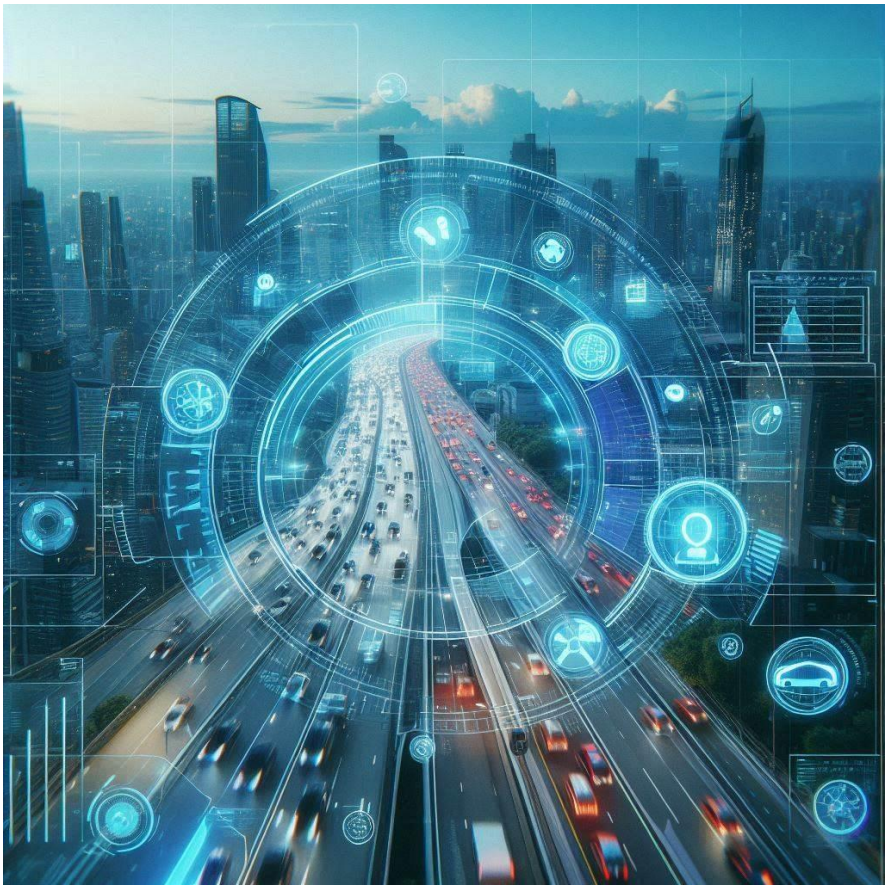
DataScientest • com

Cahier des charges projet MLOps
Accidents routiers en France



Soumaya Jendoubi Elhabibi
Marine Merle
CMLOPS-Mai24

Table des matières

1) Contexte et Objectifs

Malgré les diverses initiatives et mesures de sécurité mises en place au fil des années, le nombre d'accidents routiers restent, en France, préoccupant. En effet, en 2023, plus de 50 000 accidents corporels ont été recensés sur le territoire français, entraînant environ 3 400 décès (1). Ces chiffres mettent en lumière la nécessité de continuer à développer des solutions innovantes pour améliorer la sécurité routière et réduire le nombre et la gravité des accidents.

La Délégation à la Sécurité routière, qui a pour rôle principal de renforcer la sécurité des infrastructures routières, pourrait être le commanditaire d'un projet de machine learning visant à prédire la gravité des accidents de la route.

En exploitant des données historiques gouvernementales et en appliquant des techniques avancées de traitement et d'analyse des données, ce projet pourrait offrir des perspectives nouvelles pour les forces de l'ordre ou le SAMU pour de multiples raisons.

Ces derniers étant sur-sollicités, l'application pourrait en effet leur permettre de :

- Prioriser leurs interventions
- Gérer et optimiser les équipes à envoyer sur le lieu de l'accident
- Améliorer leur prise de décision
- Optimiser le temps de réponse en situation d'urgence.

par exemple.

L'application et ses différents micro services pourront être appelés via API, aussi bien par les forces de l'ordre que par le SAMU.

Lorsque ces derniers seront sollicités, plusieurs renseignements / paramètres seront demandés puis saisis sur l'application. A partir de ces informations, la gravité de l'accident sera prédite. Les prédictions devront être enregistrées et devront pouvoir être corrigées manuellement en cas d'erreur du modèle afin d'avoir un complément de base de données fiables sur lequel ré-entrainer le modèle.

En cas de modèle drift (ou changement de distribution dans les données) ou de model drift, le modèle devra être ré-entrainé. L'accuracy du modèle sera enregistrée. Si cette dernière est meilleure que celle du modèle précédent, le nouveau modèle devra être rendu accessible aux équipes – dans le cas contraire, l'ancien modèle sera conservé. Un système de versionning de modèle devra donc être mis en place.

Ces opérations de monitoring et maintenance seront à réaliser par les administrateurs de l'application - développeurs ou experts en data science.

2) Modèle

La variable cible que nous cherchons à prédire est la gravité de l'accident (allant de 1 à 3) : nous sommes donc dans le cas d'une tâche de classification multiple.
Les données à notre disposition d'accidents passés intègrent plusieurs variables explicatives ainsi que la gravité : il s'agit donc d'un problème supervisé.

Pour ce type de problème, plusieurs choix de modèles s'offrent à nous. Nous avons choisi d'entrainer un arbre de décision ainsi qu'une forêt aléatoire. L'arbre de décision est un modèle simple à comprendre et à interpréter, peut capturer des relations non linéaires entre les variables explicatives et la variable cible. Les modèles de forêts aléatoires permettent quant à eux de réduire les risques de sur apprentissage en améliorant la capacité de généralisation du modèle.

Nous observons grâce aux classifications report que les modèles prédisent très bien les classes de gravité 1 et moins bien les classes 3. Les classes 3 représentant les accidents les plus graves, nous cherchons à ce que notre modèle soit le plus performants dans la prédiction de cette classe.

Comparaison des résultats pour cette classe de gravité 3 :

	precision	recall	F1 score	Accuracy (toute classe)	Temps éxé
DecisionTree	0.35	0.44	0.39	0.63	1.13 sec
Random Forest	0.37	0.44	0.40	0.65	65.55 sec

La performance de ces modèles est très proche. Nous optons cependant pour le choix d'un random forest qui, bien qu'il soit plus long de quelques secondes, est un peu meilleur.

3) Base de données

Les données à notre disposition sont celles de la base de données « des accidents corporels de la circulation routière - Années de 2005 à 2021 » du
Ministère de l'Intérieur et des Outre-Mer, mise à jour le 9 octobre 2023 (2).

Ces données, réelles, sont statiques et seront stockées dans une base de données PostgreSql 'accident' et dans une table nommée 'donnees_accidents'. Le tout sera disponible dans un conteneur 'postgre_db_service'.
Dans le cas d'un projet entreprise, en réalité, les données évoluent au cours du temps dès ajout de nouvelles données – dans ce cas : à chaque intervention des forces de

l'ordre sur les lieux de l'accident.
Afin de simuler cet ajout de données continu, nous ingérerons par la suite de nouvelles données fictives.
Ces données fictives stockées dans un format .csv et disponible dans un conteneur 'db_service', seront ingérées via cron job, chaque minute, dans la base de données 'accident'.
NOTA - Afin de tester notre micro service de monitoring, nous avons fait en sorte d'intégrer un phénomène de data drift dans les nouvelles données fictives générées.
.

4) API

L'API est l'interface entre le modèle, la base de données et l'utilisateur.

Le service principal, Gateway, gèrera l'authentification des utilisateurs qui s'y connecte.
En effet, cette application ne doit être disponible que pour une liste de users définie (forces de l'ordre ou le SAMU) pour lancer des prédictions. Par ailleurs, seuls les administrateurs doivent pouvoir monitorer le modèle et le ré-entrainer. Nous devons donc gérer différents droits 'standard' ou 'admin'.
Nous avons crée une base de données avec 3 utilisateurs :

Table des utilisateurs crées			
Username	Name	Mdp	Role
User1	Sousou	Datascientest	Standard
User2	Mim	Secret	Standard
admin	admin	adminsecret	admin

Si l'utilisateur est enregistré et si ses droits lui permettent, ce dernier devra pouvoir accéder aux micro services :

- **GET / status**
  Description: Endpoint de base pour vérifier que l'API fonctionne.
  Réponse: " Bienvenue sur notre API de prédiction de la gravité des accidents routiers".

- **POST /prediction**
  Description: Endpoint pour envoyer les caractéristiques de l'accident et obtenir une prédiction sur le niveau de gravité de l'accident.
  Paramètres d'entrée: Un JSON contenant les caractéristiques routières nécessaires pour la prédiction.
  Réponse: JSON avec la prédiction de gravité.

- **PUT / correct_predict**
  Description: Endpoint pour corriger la prédiction de gravité d'un accident en cas d'erreur.
  Paramètres d'entrée: Un JSON contenant le numéro de l'accident et le niveau de gravité corrigé.
  Réponse: JSON avec un message de confirmation de changement de la gravité.

- **POST / retrain**
  Description: Endpoint pour réentrainer le modèle le modèle avec l'ensemble des données de la base.
  Paramètres d'entrée: •
  Réponse: JSON avec un message de confirmation du ré-entrainement

- **GET /monitor**
  Description: Endpoint pour recevoir l'accuracy du modèle et vérifier la présence de drift (data+model) ou non
  Paramètres d'entrée: •
  Réponse: JSON avec l'accuracy du modèle et la présence de drift ou non.

5) Testing & Monitoring

Pour garantir la fiabilité et la robustesse de l'application de prédiction des accidents routiers, les tests unitaires suivants devront être effectués :

Tests des endpoints de l'API

-Vérifier que chaque endpoint répondent correctement aux requêtes et retournent les résultats attendus. On utilise pour cela l'outil de test automatisé pytest.

Exemple :

-Tester l'endpoint "/" (status) avec des informations d'identification valides pour s'assurer qu'il répond correctement et retourne un message de bienvenue personnalisé sur l'application
-Tester l'endpoint "/" (status) avec des informations d'identification invalides pour s'assurer qu'il retourne un message d'erreur "Identifiant ou mot de passe incorrect".

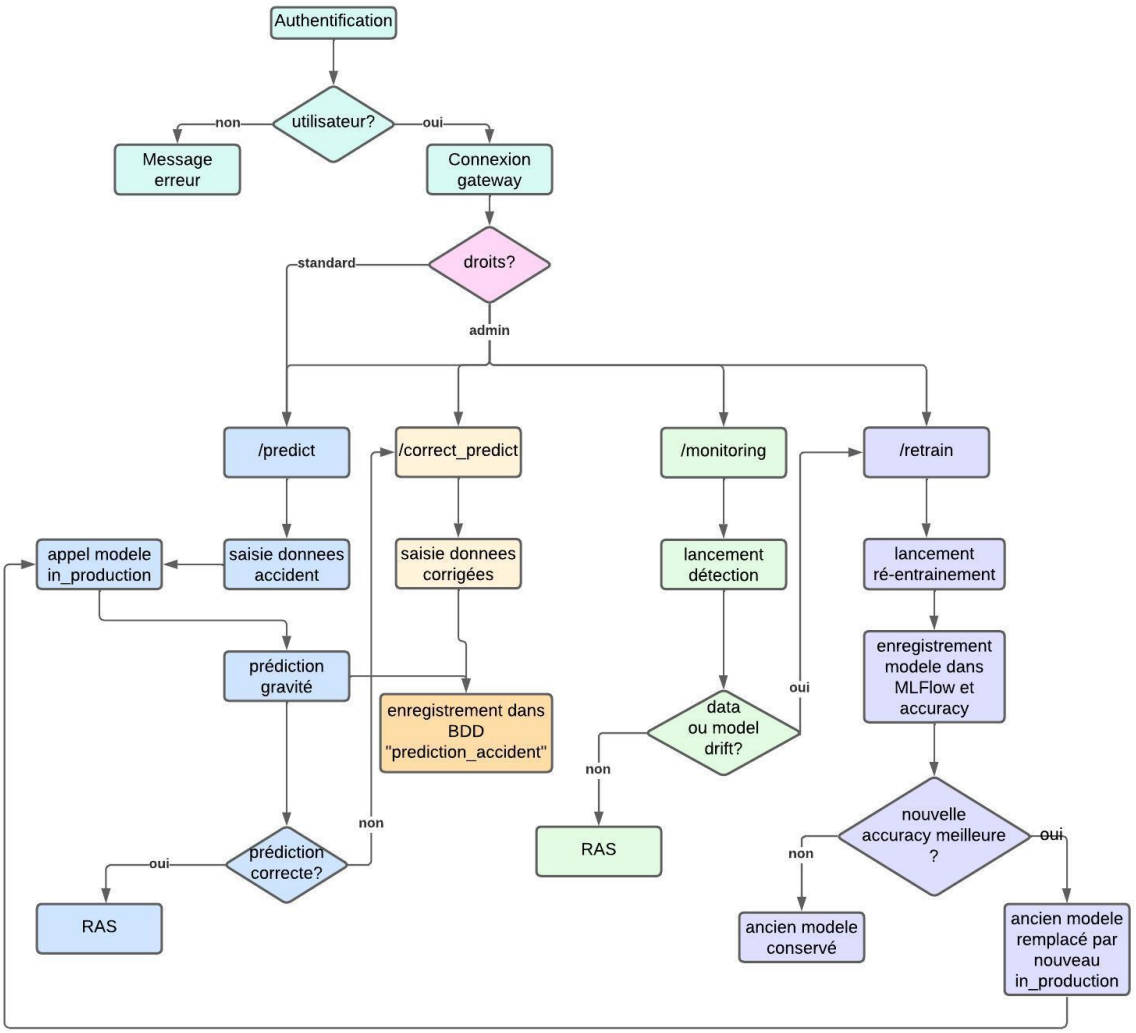
-Tester l'endpoint /prediction pour s'assurer qu'il retourne une prédiction valide pour des entrées correctes et un message d'erreur approprié pour des entrées invalides.

-Tester l'endpoint /monitoring avec des informations d'identification valides (admin) et invalides (autres users) pour s'assurer qu'il réponde correctement, vérifier que l'accuracy est supérieure à 60% et qu'il y ait ou non data / model drift

-Tester l'endpoint /retraining avec des informations d'identification valides (admin) et invalides (autres users) pour s'assurer qu'il réponde correctement, vérifier la réussite de l'entrainement

6) Schéma d'implémentation

Le schéma ci-dessous récapitule le projet mené et intègre les différentes composantes du projet et leurs interactions.



7) Sources

(1) https://www.onisr.securite-routiere.gouv.fr/etat-de-linsecurite-routiere/bilans-annuels-de-la-securite-routiere/bilan-2023-de-la-securite-routiere#:~:text=En%20France%20m%C3%A9tropolitaine%2C%203%20167,et%20254%20tu%C3%A9s%20en%202019.

(2) https://www.data.gouv.fr/fr/datasets/bases-de-donnees-annuelles-des-accidents-corporels-de-la-circulation-routiere-annees-de-2005-a-2022/