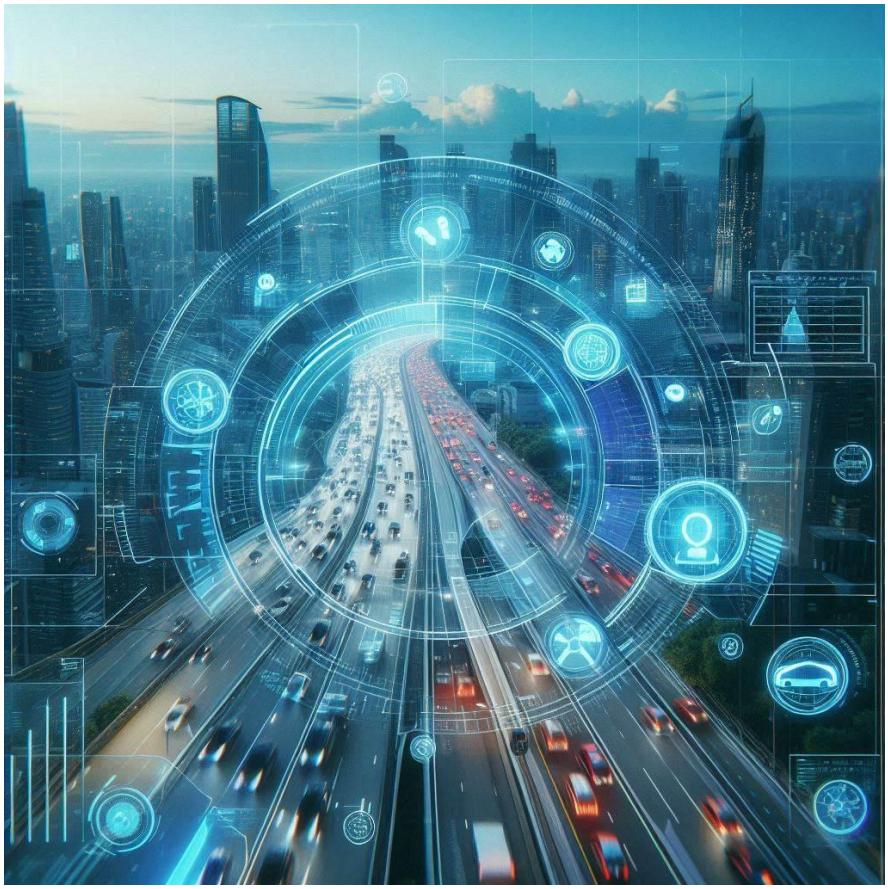
DataScientest • com

# 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.

Dans ce contexte, ce projet de machine learning visant à prédire la gravité des accidents de la route revêt une importance particulière. En exploitant des données historiques gouvernementales et en appliquant des techniques avancées de traitement et d'analyse des données, ce projet peut offrir des perspectives nouvelles pour les forces de l'ordre ou le SAMU pour de multiples raisons.

Ces derniers étant sur-sollicités, cette 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 devra ainsi être accessible via une interface web, aussi bien par les forces de l'ordre que par le SAMU.

Les informations concernant l'accident seront saisies par les forces de l'ordre, dans leurs logiciels, à chaque intervention sur les lieux de l'accident .
L'application sera connectée via API à leurs logiciels et le modèle sera ré-entrainé dès lors que ses performances ne seront plus jugées comme étant satisfaisantes.
Le commanditaire principal de l'application peut être la Délégation à la Sécurité routière qui a pour rôle principal de renforcer la sécurité des infrastructures routières.

Les administrateurs de l'application peuvent être des développeurs ou experts en data science, qui seront responsables de la maintenance, de la résolution des problèmes techniques et de l'amélioration continue de l'application.

## 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 sont donc figées. Cependant, en réalité, les données évoluent au cours du temps dès ajout de nouvelles données - à chaque intervention des forces de l'ordre sur les lieux de l'accident.
Afin de simuler cet ajout de données continu (et phénomène de data drift), nous

entrainerons donc notre modèle dans un premier temps sur les données des années 2005 à 2021 puis ingérerons par la suite les données de 2022.
.

## 4) API

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

Seules les forces de l'ordre ou le SAMU doivent pouvoir s'y connecter. Un système d'authentification doit ainsi être mis en place afin que seuls ces utilisateurs puissent s'y connecter.
Nous vérifierons ensuite leur bon accès à l'API (endpoint :'/') et l'utilisation du modèle après avoir chargé de nouvelles données concernant l'accident (endpoint : '/predict'). L'accuracy du modèle ne doit en revanche être visible que par l'administrateur de l'API (endpoint : '/accuracy') mais pas des autres utilisateurs.

Nous créons ainsi une base d'utilisateur fictive avec un rôle 'standard' et un rôle 'admin' pour tester ces différents endpoints.

```
users = {
    "user1": {
        "username": "user1",
        "name": "Sousou",
        "hashed_password": pwd_context.hash('datascientest'),
        "role": "standard",
    },
    "user2": {
        "username": "user2",
        "name": "Mim",
        "hashed_password": pwd_context.hash('secret'),
        "role": "standard",
    },
    "admin": {
        "username": "admin",
        "name": "Admin",
        "hashed_password": pwd_context.hash('adminsecret'),
        "role": "admin",
    }
}
```

Récapitulatif des endpoints à créer :

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

- **POST /predict**
  **Description**: Endpoint pour envoyer les caractéristiques routières 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.

- **GET /accuracy**
  **Description**: Endpoint pour recevoir l'accuracy du modèle actuel servant aux prédictions.
  **Paramètres d'entrée**: •
  **Réponse**: JSON avec l'accuracy du modèle.

# 5) Testing & Monitoring

## Test à effectuer

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

*Tests du modèle*

-Vérifier la précision du modèle lors de l'entraînement en s'assurant que le modèle atteigne une précision d'au moins 60%.

*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 "/" (racine) 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 "/" (racine) 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 /predict 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 /accuracy avec des informations d'identification valides (admin) et invalides (autres users) pour s'assurer qu'il répond correctement.Et vérifier que l'accuracy est supérieure à 60%.

*Tests du processus d'ingestion des données*

-Vérifier que les nouvelles données sont correctement ingérées et intégrées dans le modèle en simulant l'ajout de nouvelles données et s'assurer qu'elles sont correctement traitées et utilisées par le modèle pour de nouvelles prédictions.

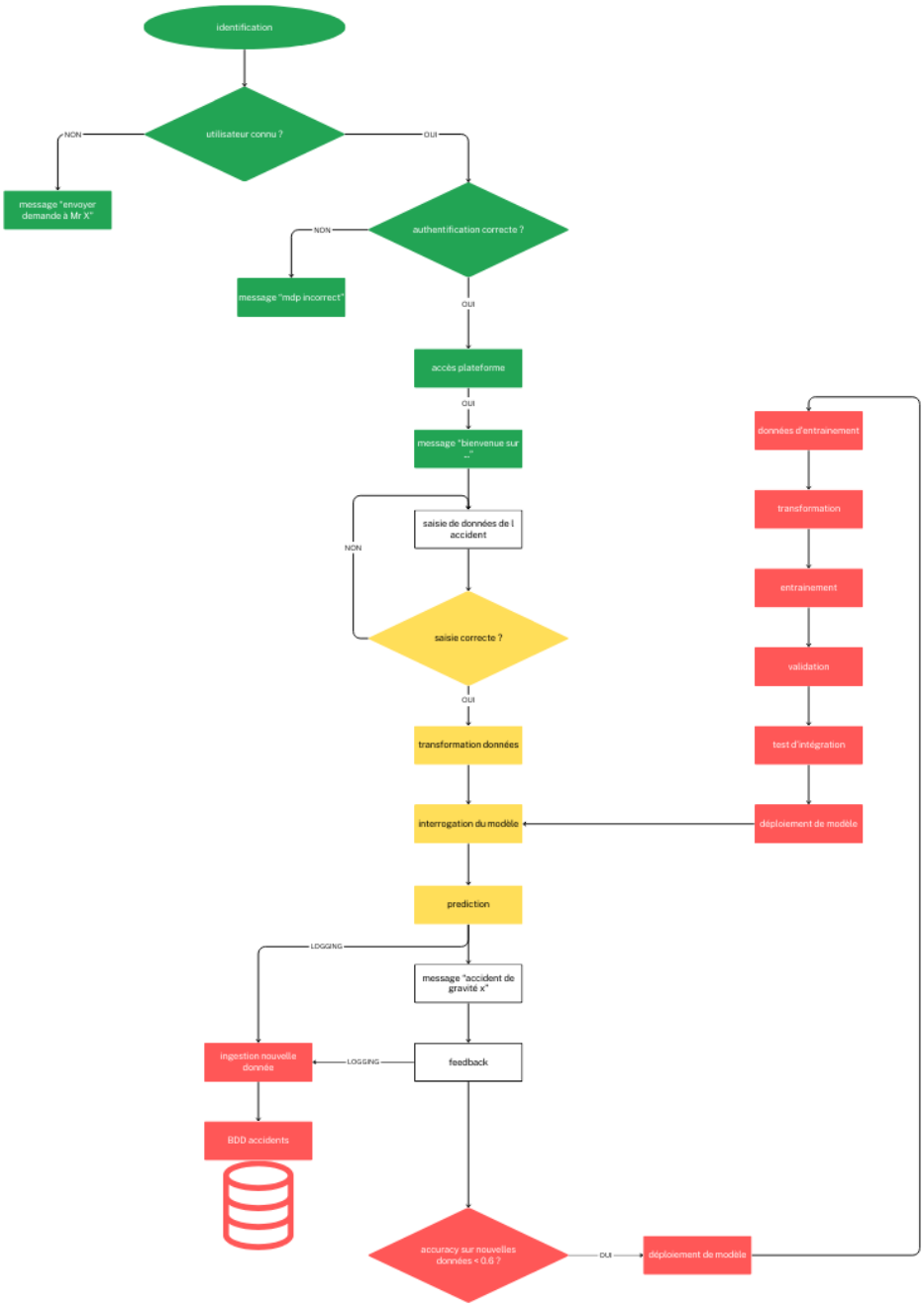
### Monitoring

Afin d'assurer le bon fonctionnement de l'application, on prévoit un ré-entrainement périodique du modèle (trimestriel) ou si la précision chute au dessous de 60%.

## 6) Schéma d'implémentation

Le schéma ci-dessous récapitule le projet mené et intègre les différentes composantes du projet et leurs interactions.
Le code couleur utilisé est le suivant :

- Accès application
- Accès données existantes et entrainement modèle
- Utilisation du modèle
- Saisie nouvelles données et monitoring




https://www.canva.com/design/DAGMhbiEcLQ/aA1ImhNJQNQqY9e4eMRogw/view?utm_content=DAGMhbiEcLQ&utm_campaign=designshare&utm_medium=link&utm_source=editor

## 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/