Prédiction de la Gravité des Accidents de la Route

Projet Machine Learning

Durée	1 semaine (5 jours)
Travail	En groupe ou individuel
Compétences	Machine Learning, ETL, API REST, Data Visualisation

Contexte du projet

Chaque année, la France enregistre environ 50 000 accidents corporels de la circulation routière. L'Observatoire National Interministériel de la Sécurité Routière (ONISR) collecte et publie ces données via les Bulletins d'Analyse des Accidents Corporels (BAAC).

Votre mission : développer un modèle capable de prédire la gravité d'un accident en fonction de ses circonstances (lieu, heure, conditions atmosphériques, type de route). Une fois le modèle entraîné, vous le déploierez via une API REST et une interface web.

Planning

Jour	Thème	Objectifs
J1	Collecte & Découverte	Télécharger les données, explorer la structure
J2	Nettoyage & ETL	Nettoyer, transformer, joindre les tables
J3	EDA & Features	Analyser, visualiser, créer des features
J4	Modélisation	Entraîner et évaluer les modèles
J5	API & Front-end	Déployer l'API et créer l'interface

Jour 1 : Collecte & Découverte

Source principale : Base BAAC

https://www.data.gouv.fr/fr/datasets/bases-de-donnees-annuelles-des-accidents-corporels-de-la-circulation-routiere-annees-de-2005-a-2024/

Fichiers à télécharger pour 2024 :

- Caract_2024.csv
- Usagers_2024.csv
- Lieux_2024.csv

Ajoutez ensuite 2 à 3 années supplémentaires pour avoir plus de volume. Téléchargez également la documentation PDF disponible sur la page.

Questions à vous poser :

- Où se trouvent les coordonnées GPS ?
- Où se trouve l'information sur la gravité des victimes ?
- Comment relier les différents fichiers entre eux ?

Jour 2 : Nettoyage & ETL

Transformez les données brutes en un dataset propre et exploitable.

Indices :

- Les coordonnées GPS ne sont pas toujours renseignées ou valides
- Certaines valeurs manquantes ne sont pas codées comme des NaN classiques
- Les informations de date/heure sont éclatées sur plusieurs colonnes
- Un accident peut impliquer plusieurs véhicules et plusieurs victimes

Jour 3 : EDA & Feature Engineering

- Analysez votre dataset : distributions, corrélations, patterns temporels et géographiques
- Visualisez les données (cartes, graphiques...)
- Créez des features pertinentes pour la prédiction
- Définissez votre variable cible (binaire ? multi-classe ?)

Jour 4 : Modélisation

- Préparez vos données pour le ML (encodage, split, gestion du déséquilibre)
- Entraînez et comparez plusieurs modèles
- Évaluez les performances (metrics adaptées)
- Analysez les features les plus importantes

Jour 5 : API & Front-end

API FastAPI

Créez une API REST avec les endpoints :

- GET /health — vérification du statut
- POST /predict — prédiction de gravité

Front-end

Créez une interface web (React, Vue, Streamlit, Gradio ou autre) permettant de saisir les caractéristiques d'un accident et d'afficher la prédiction.

Livrables

- Code source organisé (notebooks + scripts)
- Dataset nettoyé
- Modèle entraîné sauvegardé
- API fonctionnelle
- Front-end fonctionnel
- README avec instructions

Modalités d'évaluation

- Présentation orale : expliquer les choix techniques et les résultats
- Qualité du code et de la documentation
- Pertinence de l'analyse et du feature engineering
- Performance du modèle et justification des choix

Bonus (optionnel)

Pour ceux qui avancent vite, enrichissez votre dataset avec des données externes :

- Météo historique via l'API Open-Meteo (gratuite, sans clé)
- Distance aux services d'urgence (dataset data.gouv.fr)
- Temps de trajet routier via l'API OSRM

Ressources

- Documentation FastAPI : https://fastapi.tiangolo.com
- API Open-Meteo : https://open-meteo.com/en/docs/historical-weather-api
- Urgences : https://www.data.gouv.fr/fr/datasets/localisation-des-services-daccueil-des-urgences/