Prédire la gravité d'un accident de la route en France

1. Contexte

Le Ministère de l'Intérieur publie chaque année les **Bases de données annuelles des accidents corporels de la circulation routière** (fichier BAAC). Pour chaque accident, plusieurs fichiers décrivent :

Fichier	Contenu principal	Clé primaire
`caract-AAAA.csv`	Caractéristiques de l'accident (date-heure, luminosité, météo, lieu…)	`Num_Acc`
`lieux-AAAA.csv`	Infos détaillées sur l'infrastructure et l'environnement	`Num_Acc`
`usagers-AAAA.csv`	Infos par usager impliqué (âge, sexe, catégorie, sécurité…)	`Num_Acc`, `Num_Usager`
`vehicules-AAAA.csv`	Infos par véhicule (genre, motorisation, ancienneté…)	`Num_Acc`, `Num_Veh`

(AAAA = année, par ex. 2023). Lien pour y accéder :Accueil - data.gouv.fr

2. Objectifs pédagogiques

1. **Acquérir** un jeu de données 100 % ouvert et volumineux.
2. **Mener une analyse exploratoire** (EDA) pour comprendre les facteurs d'accident.
3. **Nettoyer et préparer** des données hétérogènes (dates, catégories, doublons, valeurs manquantes).
4. **Construire et évaluer** un modèle de classification pour **prédire la gravité** (« indemne », « blessé léger », « blessé hospitalisé », « tué »).
5. **Interpréter** le modèle (importance des variables, explications locales/globales).
6. **Rédiger** un notebook/documentation reproductible (Python + scikit-learn).

3. Livrables attendus

Livrable	Détails
Notebook Jupyter (ou script)	Clair, commenté, exécutable de bout-en-bout

À envoyer par email à loic.guillois+devia@react-it.fr

4. Déroulé conseillé

Étape 0 : Installation & import

Je vous recommande quelques bibliothèques:

- Python ≥3.10, pandas, numpy, matplotlib, seaborn/plotly, scikit-learn, imbalanced-learn, shap.

Étape 1 : Récupération des données

```
import pandas as pd
caract = pd.read_csv('caract-2023.csv', sep=';')
lieux  = pd.read_csv('lieux-2023.csv',  sep=';')
usagers = pd.read_csv('usagers-2023.csv', sep=';')
vehic  = pd.read_csv('vehicules-2023.csv', sep=';')
```

- Fusion initiale sur `Num_Acc` (et `Num_Usager`/`Num_Veh` si besoin).
- Conserver la cible : `grav` (gravité) dans `usagers`.

Étape 2 : Analyse exploratoire (EDA)

1. **Dimension & qualité** : nombre de lignes, de valeurs manquantes/aberrantes.
2. **Analyse univariée** : histogrammes d'âge, de vitesse limite, de luminosité.
3. **Analyse bivariée** : heatmaps / groupby pour gravité × météo, pair plot gravité × jour de la semaine, etc.
4. **Visualisation spatiale** : afficher sur une carte (latitude/longitude).
5. Identification des variables potentiellement prédictives ; documentation des hypothèses.

Étape 3 : Préparation & feature engineering

Je vous donne quelques pistes pour nettoyer et enrichir vos données.

Action	Exemple
Gestion des valeurs manquantes	Imputation médiane pour numériques, « Inconnu » pour catégorielles
Casting	`pd.Categorical` pour `lum` (luminosité), `catr` (type de route)…
Encodage	One-hot ou Target Encoding (rare levels → « Autre »)
Variables dérivées	Heure → tranche (nuit / pointe / jour), `is_weekend`, densité trafic proxy
Détection d'outliers / valeurs abbérantes	vitesses, âges

Étape 4 : Partition des données

- `train_test_split` (stratifié) : 70 % entraînement / 30 % test, ou **split temporel** (2022 → train, 2023 → test) pour éviter la fuite temporelle.
- Gestion du **déséquilibre** de classes :
  - `class_weight='balanced'` (LogReg, Tree models) ou
  - **SMOTE** / **Random Under-Sampling**.

Étape 5 : Modélisation

Voici quelques exemples de modèles que vous pouvez tester.

Modèle de base	Pourquoi
Logistic Regression	Interprétable, baseline
Random Forest	Non-linéaire, gère bien les mixtes num/
Gradient Boosting / XGBoost	Souvent la meilleure approche sur données tabulaires

LightGBM	Performance + importance gain

Évaluation avec des métriques : *accuracy*, *macro-F1*, matrice de confusion, ROC AUC (one-vs-rest).

- **Validation croisée** (K = 5) + **GridSearchCV** pour fine tuning.

Étape 6 : Interprétation & explications

- Biais potentiels (qualité des relevés, variables corrélées à l'exposition).
-

Étape 7 : Synthèse & recommandations

- Résumer les variables les plus critiques (ex. conduite de nuit, météo défavorable).
- Limites : qualité déclarative, données manquantes, causalité ≠ corrélation.
- Pistes futures : comment réduire le risque d'accident et leur gravité.