



CY TECH

MASTER II MATHÉMATIQUES APPLIQUÉES

Année universitaire 2025-2026

Use Case

Prédiction de la Gravité des Accidents Routiers

Réalisé par :

Bacarie TCHABO
Zakaria ETTOUHAMI

Encadré par :

M. MOHAMED ZERRARA

Table des matières

1	Introduction	2
2	Exploration du dataset	3
3	Feature Engineering	5
4	Modélisation	6
4.1	Random Forest	6
4.2	XGBoost	7
4.3	LightGBM	9
5	Optimisation	11
5.1	Optimisation des hyperparamètres - RandomForest	11
5.2	Optimisation des hyperparamètres - XGBoost	11
5.3	Optimisation des hyperparamètres - LightGBM	12
6	Interprétabilité : SHAP values et feature importance	14
6.1	Feature importance	14
6.2	Analyse détaillée des impacts par classe	16
7	Conclusion	18

1 Introduction

La sécurité routière représente un enjeu de santé publique majeur où la compréhension des mécanismes transformant un accident matériel en drame humain est essentielle pour orienter les politiques de prévention. L'objectif de cette étude est de développer une solution d'apprentissage supervisé capable de prédire la gravité des accidents routiers selon quatre niveaux définis par l'ONISR, allant de l'usager indemne au décès, en se basant sur les bases de données annuelles des accidents corporels des années 2022 et 2023. Le défi technique central de ce projet réside dans le fort déséquilibre des classes, les accidents mortels ne représentant qu'une infime fraction des observations (environ 2,3 %), ce qui complique considérablement leur détection par des algorithmes classiques. Pour répondre à cette problématique, notre démarche méthodologique s'est articulée autour d'une exploration approfondie des données et d'un enrichissement des variables, suivi de l'application de la méthode SMOTE pour corriger la sous-représentation des cas graves. Nous avons ensuite mis en compétition et optimisé via Optuna trois algorithmes de référence, à savoir le Random Forest, XGBoost et LightGBM, avant d'analyser les facteurs de risque déterminants grâce aux valeurs SHAP pour garantir la transparence et l'interprétabilité des décisions du modèle.

2 Exploration du dataset

Avant de procéder à la modélisation, nous avons réalisé une analyse visuelle pour comprendre la structure des données, identifier les anomalies et orienter notre stratégie de pré-traitement. Dans un premier temps, on souhaite observer le taux de valeurs manquantes pour chaque variable.

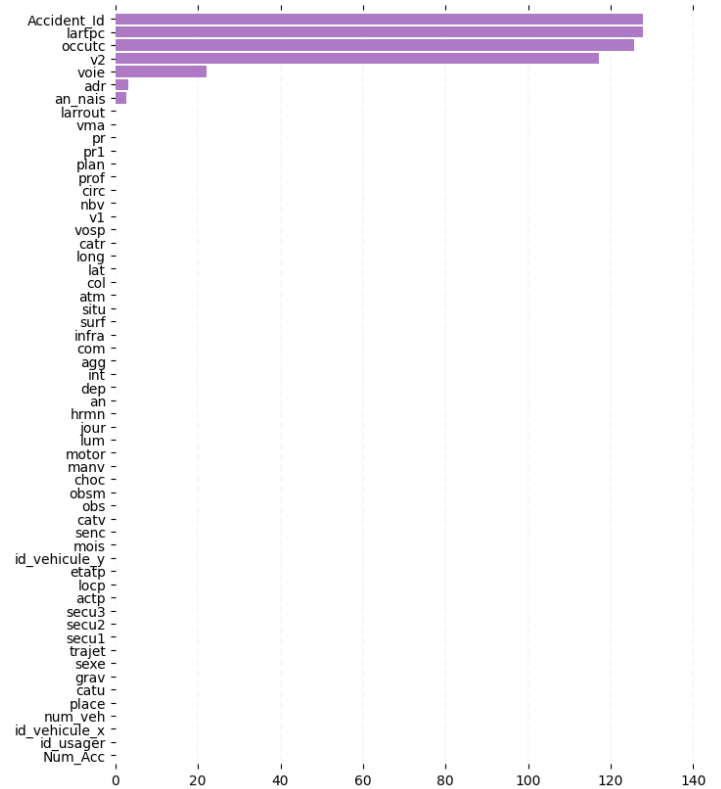


FIGURE 1 – Classement des caractéristiques avec le plus de valeurs manquantes

Le graphique met en évidence une disparité nette dans la qualité des données. La grande majorité des variables présente un taux de remplissage proche de 100%. Cependant, un groupe restreint de variables (*lartpc*, *occutc*, *v2* et *voie*), affiche un taux de valeurs manquantes extrêmement élevé. Ces variables sont trop lacunaires, elles ne contiennent pas assez d'information pour être imputées de manière fiable. Nous prenons donc le choix de les supprimer du dataset.

Ensuite, on s'intéresse à la distribution sur la variable que l'on cherche à prédire : la gravité d'un accident.

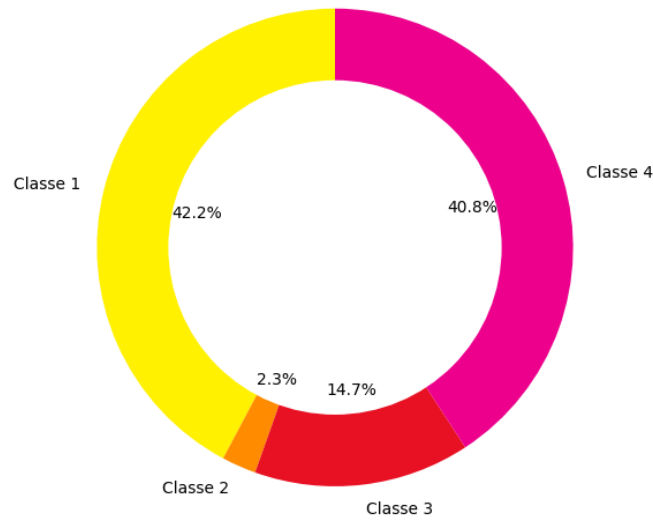


FIGURE 2 – Distribution de la variable cible (*grav*)

La variable cible *grav* souffre d'un déséquilibre de classe majeur :

- La classe 1 (indemne) et 4 (blessé léger) sont majoritaires
- La classe 3 (hospitalisé) et 2 (tué) sont minoritaires

Un déséquilibre sur la variable cible pourrait biaiser notre modèle : on pourrait obtenir un modèle extrêmement performant du simple fait que le modèle ne prédit qu'aucun accident n'est mortel. Pour palier cela, il faut utiliser une technique de rééchantillonnage. Nous utiliserons la technique SMOTE afin de générer des données synthétiques des classes minoritaires pour enrichir le jeu d'entraînement.

3 Feature Engineering

Afin d'améliorer la capacité prédictive de nos modèles, nous avons transformé les données brutes en variables explicatives pertinentes via plusieurs techniques d'enrichissement et de nettoyage.

Dans un premier temps, nous avons décidés de créer des variables temporelles. En effet, les données brutes contenaient des informations de date éclatées (année, mois, jour, heure). Nous les avons retravaillées pour capturer des cycles comportementaux liés aux accidents :

- L'heure de l'accident a été extraite et discrétisée via la variable *hrmn*. Nous avons transformé la variable continue en 24 classes horaires (allant de 00h à 23h). Cela permet au modèle de détecter les pics d'accidents liés aux heures de pointe ou aux sorties nocturnes.
- Nous avons reconstitué une date complète pour déterminer le jour de la semaine. Une variable binaire *is_weekend* a été créée (1 pour Samedi/Dimanche, 0 sinon). Cette feature est cruciale car la typologie et la gravité des accidents diffèrent souvent entre la semaine (trajets domicile-travail) et le week-end (loisirs, alcoolémie potentielle).

Dans un second temps, nous nous sommes intéressées aux données géographiques du dataset. Nous disposons de la latitude et de la longitude de l'endroit de l'accident, respectivement via les variables *lat* et *long*. Cependant, les coordonnées GPS brutes sont trop granulaires et difficiles à interpréter pour un modèle d'arbre de décision. Pour capturer l'effet "zone géographique" sans utiliser les milliers de codes communes. Nous avons donc appliqué un algorithme K-Means sur les coordonnées géographiques : nous avons segmenté le territoire en 15 clusters distincts. Cela permet de regrouper les accidents par "zones à risque" homogènes géographiquement, simplifiant ainsi l'information spatiale pour le modèle.

Bien évidemment, l'utilisation d'algorithmes de machine learning implique un encodage des variables catégorielles :

- Nous appliquons un Ordinal Encoding à la variable cible *grav* (pour la transformer en classes 0, 1, 2, 3) et à certaines variables d'identification ou ordonnées.
- Nous appliquons un One-Hot Encoding aux variables catégorielles nominales (comme *catv* pour la catégorie de véhicule, *lum* pour la luminosité, etc.).

Enfin, comme identifié lors de l'exploration, la classe "tué" est très minoritaire au sein de la variable cible. Pour empêcher le modèle de biaiser ses prédictions vers la classe majoritaire, nous avons utilisé la technique SMOTE (Synthetic Minority Over-sampling Technique).

Le SMOTE génère des exemples synthétiques des classes minoritaires en interpolant entre des points existants proches.

4 Modélisation

Dans cette partie, nous affichons les résultats issues de la modélisation par les différents modèles utilisés (Random Forest, XGBoost, LightGBM).

4.1 Random Forest

Le premier modèle testé est un Random Forest Classifier, entraîné sur les données rééquilibrées par SMOTE. Voici l'analyse de ses performances sur le jeu de test.

Modèle	RandomForest
précision	74.7%
F1-score	74.1%

TABLE 1 – Résultats du modèle RandomForest

De la précision, on conclut que le modèle classe correctement près de 3 accidents sur 4. Quant au F1-Score, qui pondère la précision et le rappel selon le volume de chaque classe, sa valeur (74%) est cohérente avec l'accuracy, suggérant une performance globale stable. Cependant, cette performance globale masque des disparités importantes entre les classes, comme le montre le rapport de classification détaillé :

Classe	précision	recall	f1-score	support
1	0.79	0.86	0.82	13186
2	0.44	0.18	0.26	730
3	0.59	0.55	0.57	4580
4	0.75	0.73	0.74	12736

TABLE 2 – rapport de classification détaillé - RandomForest

On en déduit que le modèle est performant sur classes majoritaires (classe 1 et 4) et très peu performant sur les classes minoritaires (classe 2 et 3). Le plus critique est la performance sur la classe 2, c'est clairement le point faible majeur du modèle. Avec un rappel de seulement 0.18, le modèle ne détecte que 18% des accidents mortels. La précision est faible (0.44), indiquant de nombreuses fausses alertes.

La matrice de confusion confirme visuellement ces difficultés de distinction entre les niveaux de gravité adjacents :

- Notre modèle a tendance à confondre les personnes "tué" et les personnes "hospitalisé" : Sur les 730 cas réels de décès (ligne 1), le modèle en classe 340 comme "hospitalisé" (colonne 2) et 179 comme "léger" (colonne 3). Il a tendance à sous-estimer la gravité des accidents mortels.
- Notre modèle a tendance à confondre les personnes "hospitalisé" et les personnes dite "blessé léger" : Sur les accidents nécessitant une hospitalisation (ligne 2), une grande partie (1758 cas) est prédite à tort comme "blessé léger".

Pour conclure, bien que le Random Forest offre une stabilité globale correcte (75%), il échoue à remplir l'objectif prioritaire du projet : identifier efficacement les accidents les plus graves (classe 2). Malgré l'utilisation de SMOTE, les caractéristiques des accidents mortels et hospitalisés semblent trop proches pour être distinguées linéairement par ce

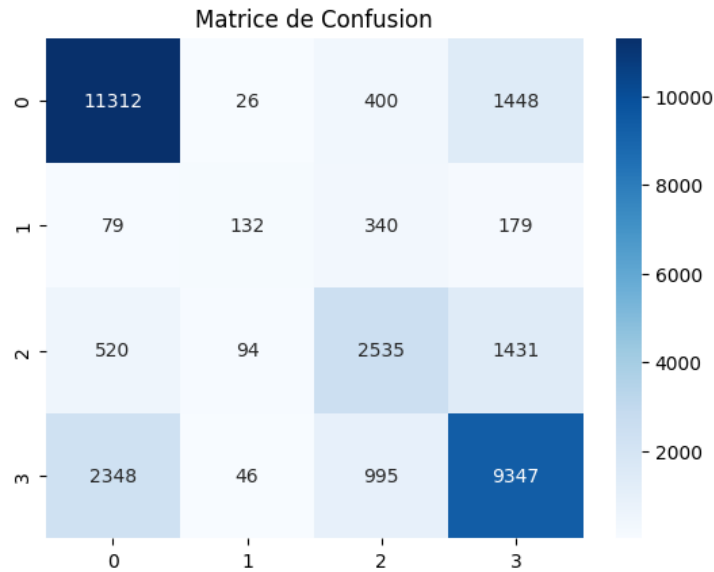


FIGURE 3 – Matrice de corrélation - RandomForest

modèle. Cela justifie le test de modèles de boosting (XGBoost, LightGBM) pour tenter de mieux capturer ces subtilités.

4.2 XGBoost

Le second modèle évalué est le XGBoost.

Modèle	XGBoost
précision	67.5%
F1-score	66.7%

TABLE 3 – Résultats du modèle XGBoost

D'un point de vue précision, on observe une baisse significative de 7 points par rapport au Random Forest. Quant au F1-Score, la cohérence globale des prédictions diminue elle aussi. Si l'on regarde en détail le comportement de notre modèle :

Classe	précision	recall	f1-score	support
1	0.73	0.84	0.78	13186
2	0.22	0.18	0.20	730
3	0.50	0.39	0.44	4580
4	0.68	0.63	0.66	12736

TABLE 4 – rapport de classification détaillé - XGBoost

En ce qui concerne les classes dites "critiques" (classe 2 et 3) :

- Pour la classe 2, on observe un phénomène intéressant. Le rappel monte à 0.20 (contre 0.19 pour le Random Forest). Le modèle réussit donc à détecter davantage d'accidents mortels. Cependant, cette amélioration se paie cher : la précision s'effondre à 0.18. Cela signifie que lorsque le modèle prédit "tué", il se trompe plus de 7 fois sur 10. Il génère énormément de "bruit".

- Pour la classe 3 : La performance se dégrade nettement avec un F1-score de 0.44. Le modèle confond massivement ces blessés graves avec des blessés légers ou indemnes.

En ce qui concerne les classes majoritaires (classe 1 et 4), contrairement au Random Forest, XGBoost peine davantage à segmenter correctement même les cas simples. Le rappel de la classe 4 chute à 0.63, indiquant que beaucoup de blessés légers sont mal classifiés.

De plus, la matrice de confusion révèle une dispersion des prédictions beaucoup plus "floue" que pour le modèle précédent :

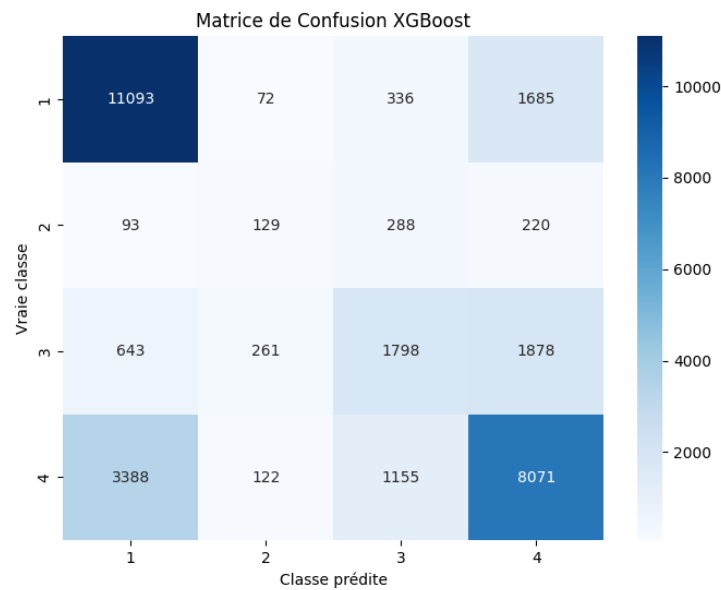


FIGURE 4 – Matrice de corrélation - XGBoost

On observe une explosion de faux positifs sur la classe 2 : en regardant la colonne 2 (prédictions "classe 2"), on voit que le modèle a classé comme "tués" un grand nombre de blessés hospitalisés (261 cas) et même de blessés légers (122 cas). C'est ce qui explique la précision très faible. En outre, il y a une confusion entre les classes hospitalisé et indemne : Pour la vraie classe 3 (ligne 3), le modèle disperse ses prédictions : 1798 sont corrects, mais 1878 sont prédits comme légers et 576 comme indemnes. Cette sous-estimation de la gravité est problématique. Enfin, Une erreur massive apparaît en bas à gauche : 3388 blessés légers (ligne 4) sont prédits comme indemnes (colonne 1), ce qui est beaucoup plus élevé que dans le modèle Random Forest.

Ainsi, Dans cette première itération, XGBoost se montre plus "agressif" sur la classe minoritaire, ce qui augmente mécaniquement le nombre de décès détectés (rappel), mais il perd en stabilité globale. La chute de l'accuracy et la précision médiocre sur les classes critiques suggèrent que le modèle a du mal à généraliser et qu'il est peut-être perturbé par le bruit introduit par le SMOTE ou un réglage d'hyperparamètres (learning rate, profondeur) inadapté. Une optimisation via Optuna est indispensable pour tenter de canaliser cette agressivité et réduire les fausses alertes.

4.3 LightGBM

Le dernier modèle testé est le LightGBM. Ses performances se situent à mi-chemin entre le Random Forest et le XGBoost.

Modèle	LightGBM
précision	68.2%
F1-score	67.3%

TABLE 5 – Résultats du modèle LightGBM

Le modèle fait légèrement mieux que XGBoost (+1.3 points), mais reste nettement en retrait par rapport au Random Forest. Analysons en détail le comportement du modèle :

Classe	précision	recall	f1-score	support
1	0.74	0.84	0.78	13186
2	0.25	0.16	0.19	730
3	0.51	0.40	0.45	4580
4	0.68	0.66	0.67	12736

TABLE 6 – rapport de classification détaillé - LightGBM

En ce qui concerne les classes minoritaires (classe 2 et 3) :

- Pour la classe 2, les résultats montrent une tentative de rééquilibrage par rapport à XGBoost. La précision remonte à 0.25 (contre 0.22 pour XGBoost), ce qui signifie que le modèle génère moins de fausses alertes. Cependant, le Rappel chute à 0.19, un niveau similaire à celui du Random Forest. Le modèle est donc "plus prudent" que XGBoost mais rate la grande majorité des accidents mortels (plus de 80%).
- Pour la classe 3, avec un rappel de 0.45, LightGBM peine à identifier les blessés hospitalisés, performant moins bien que le Random Forest (0.51) sur ce segment crucial.

En ce qui concerne les classes majoritaires, le modèle assure l'essentiel sur les indemnes (rappel = 0.83) mais montre des faiblesses sur les blessés légers (rappel = 0.63), les confondant souvent avec des indemnes.

Si l'on regarde la matrice de confusion :

L'examen de la matrice de confusion du LightGBM met en lumière une tendance marquée à la sous-estimation de la gravité. Dans un premier temps, le modèle a tendance à sous estimer la gravité d'un accident : sur les 730 décès réels (Ligne 2), seuls 133 sont correctement prédits. Le modèle classe la majorité des décès comme "blessé léger" (277 cas) ou "hospitalisé" (239 cas). De plus, le modèle a tendance à confondre les cas où l'accident mène à une hospitalisation et les cas où l'accident mène à un blessé léger. C'est l'erreur la plus massive du modèle : sur la ligne 3 (hospitalisés), 2185 victimes sont prédites à tort comme "blessés légers". Le modèle minimise la gravité de l'accident dans près de la moitié des cas d'hospitalisation. De même, 3271 blessés légers sont classés comme indemnes (ligne 4, colonne 1), ce qui constitue un risque de non-prise en charge dans un contexte opérationnel.

Pour conclure, LightGBM offre une performance plus "propre" que le XGBoost en termes de précision pure sur les classes critiques, générant moins de bruit. Cependant, il

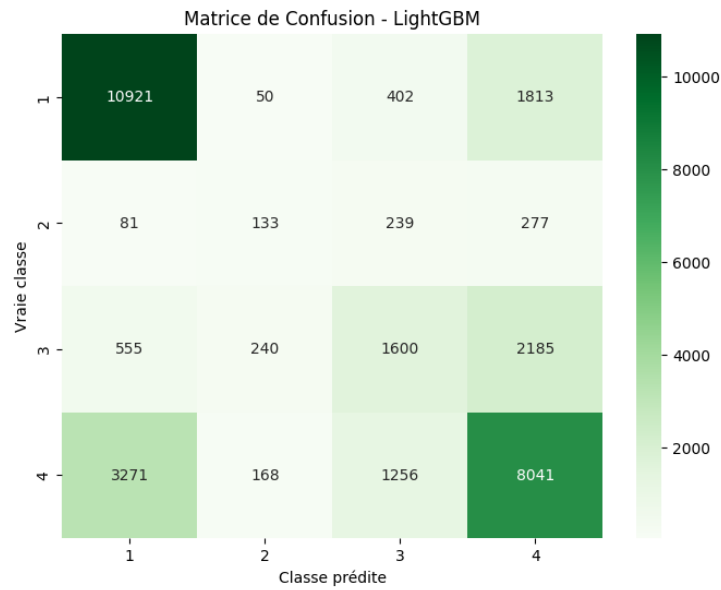


FIGURE 5 – Matrice de corrélation - LightGBM

souffre du même défaut structurel que les autres modèles : une difficulté majeure à distinguer les nuances de gravité intermédiaires (Hospitalisé vs Léger). La sous-estimation systématique de la gravité (tendance à prédire la classe inférieure) suggère que les caractéristiques distinctives des accidents graves ne sont pas suffisamment captées par les variables actuelles, malgré le SMOTE.

5 Optimisation

Dans cette partie, nous cherchons à optimiser les hyperparamètres des modèles que nous avons utilisés.

5.1 Optimisation des hyperparamètres - RandomForest

Après optimisation des hyperparamètres, nous obtenons les résultats suivants :

Modèle	RandomForest	RandomForest optimisé
précision	74.7%	72.5%
F1-score	74.1%	71.9%

TABLE 7 – Résultats du modèle RandomForest et du modèle RandomForest optimisé

Cette baisse de performance du modèle après optimisation des hyperparamètres s'explique par la différence d'hyperparamètres :

Modèle	RandomForest	RandomForest optimisé
min_samples_leaf	1	1
min_samples_split	2	11
max_depth	None	35

TABLE 8 – Comparaison des hyperparamètre entre le modèle de base et le modèle optimisé

Les paramètres optimisés imposent des contraintes plus fortes. Il modèle est plus généraliste, mais il a perdu la capacité de "par cœur" du modèle par défaut sur les cas faciles, ce qui fait baisser l'accuracy globale.

De plus, la contrainte métier nous impose de maximiser le F1-score. pour gagner un tout petit peu de performance sur les classes minoritaires, le modèle sacrifie beaucoup de performance sur les classes majoritaires. Comme les individus des classes majoritaires sont très nombreux dans le dataset, une petite baisse de précision sur ces classe fait chuter le score global drastiquement.

Néanmoins, on peut arguer que le modèle optimisé est meilleure bien qu'il soit moins performant que le modèle de base :

5.2 Optimisation des hyperparamètres - XGBoost

L'optimisation des hyperparamètres via Optuna a permis de stabiliser considérablement le modèle XGBoost par rapport à sa version initiale. La recherche a convergé vers une architecture relativement complexe et robuste.

Modèle	XGBoost	XGBoost optimisé
précision	64.7%	75.2%
F1-score	64.2%	74.6%

TABLE 9 – Résultats du modèle XGBoost et du modèle XGBoost optimisé

L'optimisation a apporté un gain de performance notable, on gagne plus de 10 points de précision et plus de 10 points d'exactitude, corrigeant la dispersion "bruitée" observée sur la version précédente.

Si l'on observe les résultats détaillés par classes :

Classe	précision	recall	f1-score	support
1	0.80	0.86	0.83	13186
2	0.49	0.21	0.29	730
3	0.62	0.53	0.57	4580
4	0.74	0.76	0.75	12736

TABLE 10 – rapport de classification détaillé - XGBoost optimisé

Le modèle de base nous donnait de bons scores de rappel mais une précision catastrophique sur les décès. Ici, la version optimisé du modèle a changé de stratégie :

- Pour la classe 2 (tué), on obtient un gain de fiabilité : La précision a presque triplé, passant de 0.16 à 0.49. Désormais, quand le modèle prédit un décès, il a raison près d'une fois sur deux (contre une fois sur six auparavant). Cela réduit drastiquement les fausses alertes.
- Cependant, en devenant plus précis, le modèle est devenu plus "prudent". Le rappel de la classe 2 est descendu à 0.29. Il rate donc davantage de décès réels.

Pour conclure, l'étape d'optimisation a transformé un modèle instable en un classifieur fiable sur les classes majoritaires (F1-score > 0.80 pour les indemnes) et beaucoup plus crédible sur les classes critiques. Bien que le rappel sur les décès reste faible, la forte augmentation de la précision rend les alertes de ce modèle beaucoup plus exploitables pour une prise de décision, car elles génèrent moins de bruit.

5.3 Optimisation des hyperparamètres - LightGBM

L'optimisation des hyperparamètres a permis au modèle LightGBM de rattraper son retard sur le Random Forest et d'égaliser les performances du XGBoost optimisé, tout en affichant une fiabilité accrue sur les prédictions critiques.

Modèle	LightGBM	LightGBM optimisé
précision	68.2%	72.5%
F1-score	67.3%	72.0%

TABLE 11 – Résultats du modèle LightGBM et du modèle lightGBM optimisé

Comme pour XGBoost, le gain après optimisation est significatif. On gagne un peu plus de 4 points sur la précision et un peu plus de 5 points pour le F1 score. Globalement, le modèle se hisse au même niveau de performance générale que le XGBoost optimisé.

Si l'on observe les métriques par classe, on remarque que c'est sur la gestion de la classe minoritaire que ce modèle se distingue le plus :

Classe	précision	recall	f1-score	support
1	0.78	0.84	0.81	13186
2	0.42	0.19	0.26	730
3	0.58	0.50	0.53	4580
4	0.71	0.72	0.72	12736

TABLE 12 – rapport de classification détaillé - LightGBM optimisé

- En ce qui concerne la classe 2, le LightGBM optimisé obtient une précision de 0.48. C'est le score le plus élevé parmi tous les modèles testés.
- En ce qui concerne les classes majoritaires, le modèle maintient d'excellentes performances sur les indemnes (F1-score 0.80) et les blessés légers (F1-score 0.70), assurant la stabilité du modèle.

Pour conclure, le LightGBM optimisé s'impose probablement comme le modèle le plus équilibré du projet. Bien que son accuracy globale soit infiniment inférieure à celle de XGBoost (70.86% contre 75.2%), il offre le meilleur compromis précision/rappel sur la classe critique des décès.

6 Interprétabilité : SHAP values et feature importance

6.1 Feature importance

Pour comprendre sur quels critères le modèle LightGBM optimisé se base pour prédire la gravité, nous avons analysé le "gain d'information" de chaque variable. Le graphique ci-dessous présente les 20 variables les plus influentes.

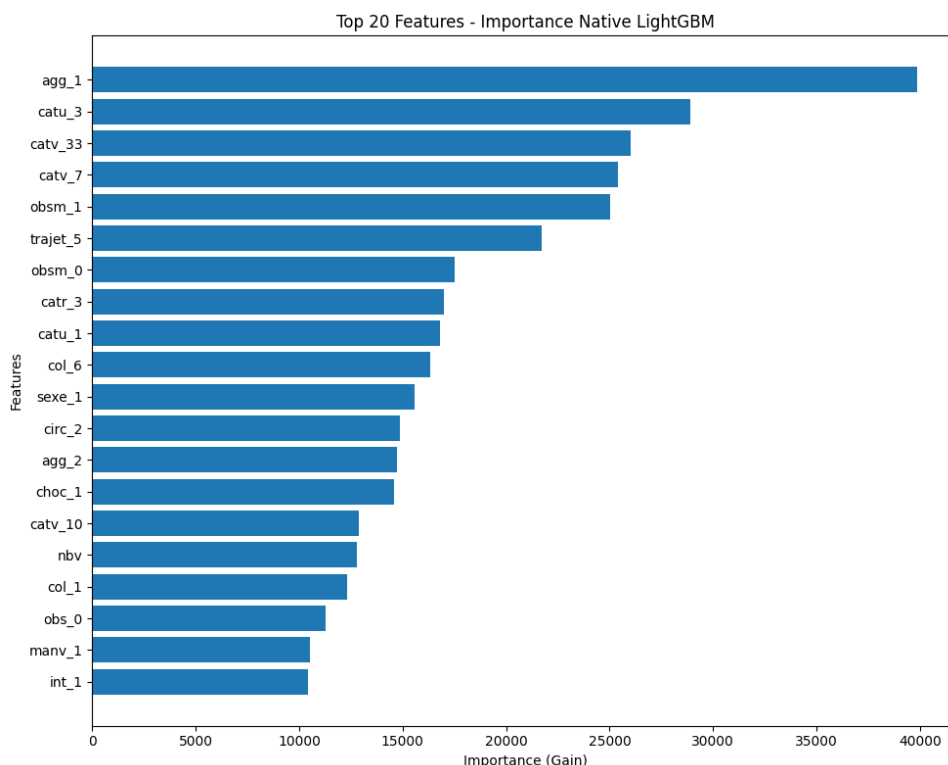


FIGURE 6 – Les 20 variables les plus importantes pour le modèle LightGBM optimisé

La variable *agg_1* (correspondant à la localisation "hors agglomération" vs "en agglomération") domine écrasante le classement avec un gain de près de 40 000. C'est le discriminateur principal de la gravité. Cela s'explique par la réglementation : la localisation détermine la limitation de vitesse (50 km/h en ville vs 80/110/130 km/h hors agglomération). De ce fait, les accidents hors agglomération sont plus violents et donc plus souvent mortels.

Parmi les variables importantes, on retrouve *obsm_1* (obstacle mobile heurté = piéton) qui arrive en 5ème position et *catu_3* (catégorie usager = Piéton) qui arrive en 2ème position. Les piétons jouent un rôle central dans la décision du modèle : le modèle a correctement identifié que l'implication d'un piéton est un indicateur critique de gravité élevée.

La variable *catv_33* (motocyclettes lourdes > 125 cm³) se place sur le podium. Cela reflète la sur-représentation des motards dans les accidents graves et mortels.

Le motif du déplacement "promenade / loisirs" (*trajet_5*) apparaît en 6ème position, bien avant les trajets domicile-travail. Ce type de trajet est souvent associé aux week-ends,

à des routes méconnues, voire à une consommation d'alcool ou à la fatigue, augmentant le risque d'accidents sévères.

Enfin, les routes départementales () et certains types de collision frontale ou latérale (*col_6*) sont également des vecteurs importants d'information pour le modèle. Les départementales, souvent bidirectionnelles sans séparateur central et bordées d'obstacles (arbres), sont les routes les plus meurtrières en France.

Pour conclure, la hiérarchie des variables établie par le modèle est cohérente avec la réalité de la sécurité routière. Le modèle ne se base pas sur des artefacts statistiques (bruit) mais bien sur les fondamentaux physiques du risque : la vitesse , la vulnérabilité (piéton, moto) et l'environnement.

6.2 Analyse détaillée des impacts par classe

Le graphique des valeurs SHAP moyennes permet d'aller plus loin que la simple "feature Importance" en décomposant l'impact de chaque variable pour chacune des 4 classes de gravité :

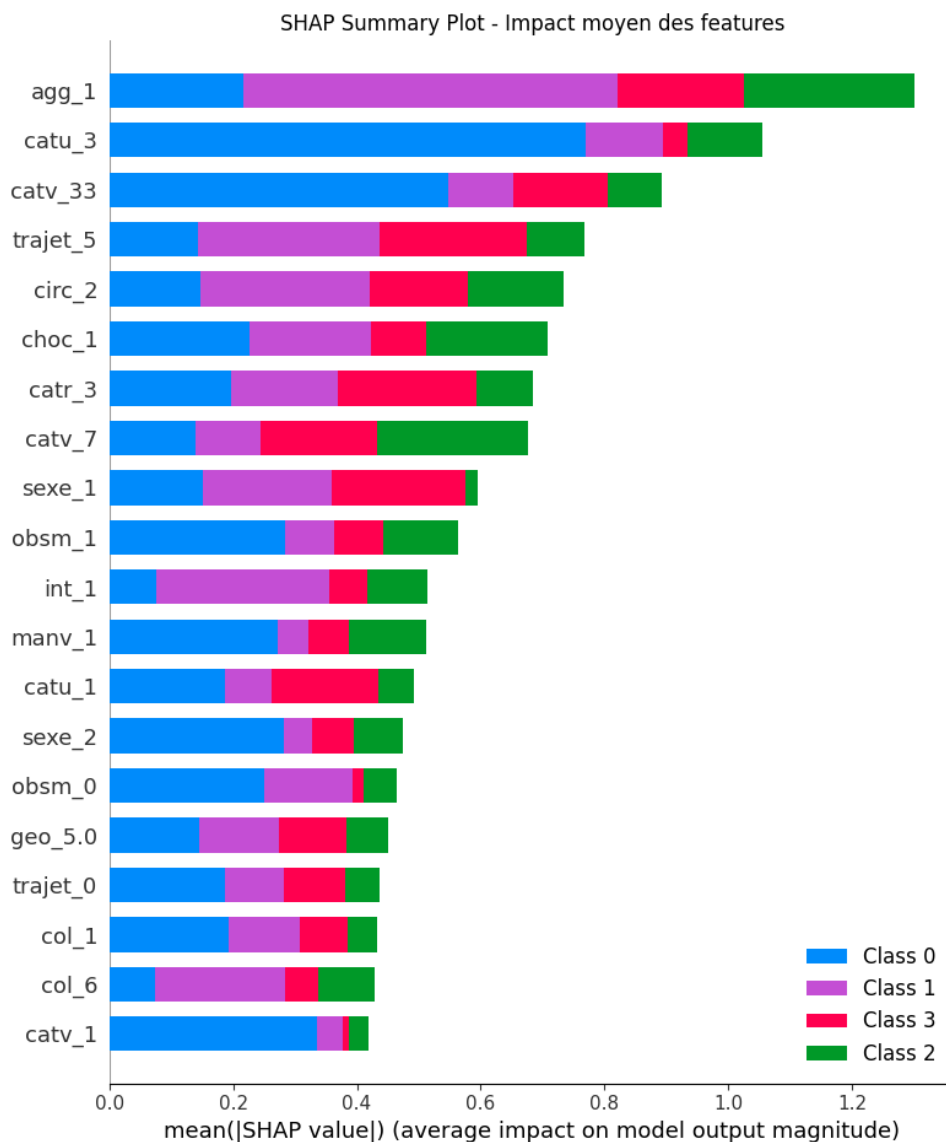


FIGURE 7 – Classement de l'impact moyen des variables - détail par classe

Dans un premier temps, on observe que la localisation joue un rôle pivot. En effet, La variable *agg_1* (hors agglomération) est la plus influente du modèle, mais le graphique révèle qu'elle pèse principalement sur les classes indemne (bleu) et blessé léger (rouge). Le modèle utilise la localisation avant tout pour "trier" la masse des accidents peu graves. En agglomération, les accidents sont majoritairement matériels ou légers. Hors agglomération, le risque bascule. C'est le premier filtre du modèle.

Ensuite, pour détecter les accidents mortels, le modèle s'appuie spécifiquement sur certaines variables où la portion violette est particulièrement visible :

- *catv_33* (moto lourde > 125cm³) : C'est la variable qui possède proportionnellement l'un des impacts les plus forts sur la classe "tué". La présence d'une moto lourde est un signal d'alarme majeur pour le modèle concernant le risque de décès.
- *catu_3* (piéton) : De même, cette variable active fortement la prédiction des classes graves (Sections violetes pour tué, sections vertes pour hospitalisé).
- *catr_3* (route départementale) : On observe une contribution notable à la classe "tué", confirmant la dangerosité spécifique de ce type de route (bidirectionnel, vitesse 80km/h, etc...).

Enfin, on voit que la difficulté du modèle à séparer les hospitalisés (sections vertes) des blessés légers (sections rouges) se lit ici. Pour beaucoup de variables contextuelles comme *trajet_5* (loisirs) ou *circ_2* (bidirectionnel), les contributions aux classes "blessé léger" et "hospitalisé" sont d'ampleur comparable. Cela explique pourquoi le modèle confond souvent ces deux classes : les contextes sont similaires, la différence se joue souvent à des détails (port de la ceinture, angle de choc, etc...) que le modèle peine parfois à capter.

7 Conclusion

Au terme de cette étude, nous avons réussi à mettre en place une chaîne de traitement permettant de prédire la gravité des accidents avec une fiabilité opérationnelle, confirmant par l'analyse exploratoire la nécessité absolue de traiter le déséquilibre des classes pour ne pas noyer les accidents mortels dans la masse des accidents légers. La comparaison des modèles a mis en évidence une progression nette des performances, où le Random Forest a montré ses limites dans la détection des cas graves contrairement aux modèles de boosting optimisés. Le LightGBM s'est finalement imposé comme le modèle le plus équilibré, offrant le meilleur compromis pour isoler les profils à risque avec une précision accrue sur la classe des décès. L'analyse d'interprétabilité a par ailleurs validé la pertinence physique du modèle en redécouvrant les fondamentaux de la sécurité routière, identifiant l'énergie cinétique et la vulnérabilité des usagers comme les facteurs discriminants principaux.