



DataScientest • com

Décembre 2023 – Juillet 2024

Accidents routiers en France

Projet mené par :
Matthieu Claudel
Vanessa Ibert
Camille Pelat
Nadège Reboul

Sommaire

INTRODUCTION AU PROJET	8
I.1 Contexte	8
I.2 Objectifs.....	9
I COMPREHENSION ET MANIPULATION DES DONNEES	11
I.1 Cadre.....	11
I.2 Pertinence	13
I.3 Pre-processing et feature engineering	14
I.4 Exemples détaillés avec visualisation	20
I.5 Analyse des variables continues	29
I.5.1 Age_usager.....	30
I.5.2 Heure et mois.....	30
I.5.3 Latitude et longitude	30
I.6 Visualisations et Statistiques.....	31
I.6.1 Evolution Temporelle.....	31
I.6.2 Disparités spatiales	36
I.6.3 Influence de l'âge et du sexe dans l'accidentalité routière	37
I.7 Conclusion	39
I.8 Perspectives de modélisation	40
II PRÉDICTION DU NOMBRE D'INDEMNIES, DE BLESSÉS LÉGERS, DE BLESSÉS HOSPITALISÉS ET DE TUÉS PAR JOUR	42
II.1 Séries temporelles	42
II.1.1 Création des jeux de données	42
II.1.2 Etude de la saisonnalité.....	43
II.1.3 Création d'une baseline	45
II.1.4 Prédictions.....	50
II.1.5 Conclusion	62
III PRÉDICTION DE LA GRAVITÉ DE L'ACCIDENT – MACHINE LEARNING	67
III.1 Régression logistique.....	67
III.1.1 Modèle de référence	67
III.1.2 Optimisation du modèle	68
III.1.3 Interprétabilité des résultats	69
III.2 Support Vector Machine.....	72
III.2.1 Modèle de référence	72
III.2.2 Optimisation du modèle - Ajustement des hyperparamètres penalty, loss, C et multi_class avec Optuna	72

III.3	Decision Tree Classifier - Random Forest Classifier - Balanced Random Forest..	73
III.3.1	Modèle de référence	73
III.3.2	Optimisation du modèle	75
III.3.3	Résultats.....	76
III.3.4	Interprétabilité des résultats	78
III.4	CatBoost Classifier	81
III.4.1	Modèle de référence	81
III.4.2	Optimisation du modèle	82
III.4.3	Interprétabilité des résultats	83
III.5	XGBoost Classifier.....	85
III.5.1	Modèle de référence	85
III.5.2	Optimisation de la méthode 2	89
III.5.3	Interprétabilité des résultats avec SHAP	90
III.6	K-nearest neighbors (KNN).....	93
III.6.1	Modèle de référence	93
III.6.2	Recherche de paramètres optimaux avec GridSearchCV	94
III.6.3	Sélection des features	96
IV	ESSAIS D'OPTIMISATION DE LA PRÉDICTION DE LA GRAVITÉ DE L'ACCIDENT – MACHINE LEARNING	97
IV.1	Classification binaire	97
IV.1.1	Modélisation	97
IV.1.2	Comparaison des résultats avec la classification multilabels	98
IV.1.3	Interprétabilité des résultats	99
IV.2	Etude des variables ‘catv’, ‘obs’ et ‘obsm’	105
IV.2.1	Modifications des variables	105
IV.2.2	Modélisation	105
IV.2.3	Résultats et comparaison avec le jeu de données initial	106
IV.3	Conclusion des modélisations de Machine Learning.....	107
V	PREDICTION DE LA GRAVITE DE L'ACCIDENT – DEEP LEARNING	110
V.1	Modélisation par Deep Learning avec Keras.....	110
V.1.1	Modèle de référence	110
V.1.2	Rééquilibrage du jeu de données.....	111
V.1.3	Hyperparamétrage du modèle	113
V.2	Modélisation par Deep Learning avec PyTorch	115
V.2.1	Modèle de référence	115
V.2.2	Hyperparamétrage	116
V.3	Modélisation par Deep Learning avec TabNet.....	116

V.3.1	Optimisation des hyperparamètres avec Optuna	117
V.3.2	Interprétabilité du modèle	118
V.4	Comparaison Deep Learning / Machine Learning (classification multi-classes)...	118
VI	RECENTS ESSAIS D'OPTIMISATION – AJOUT D'UNE NOUVELLE VARIABLE	119
VI.1	Création de la variable ‘nb_usagers_gr’	119
VI.2	Modélisation.....	119
VI.3	Résultats et comparaison avec le jeu de données initial	119
VII	CONCLUSIONS ET PERSPECTIVES	120
VII.1	Objectifs atteints.....	120
VII.2	Difficultés rencontrées lors du projet.....	123
VII.3	Bilan	125
VII.4	Perspectives.....	126
	BIBLIOGRAPHIE	127

Table des Figures

Figure 1 : Liste des variables présentes dans les bases de données mises à disposition du grand public	12
Figure 2 (a). Nombre d'accidents par an, (b). Nombre de tués par an	13
Figure 3 : Pourcentage d'accidents selon la modalité pour les années 2019 à 2022	14
Figure 4 : Pourcentage de valeurs manquantes pour chaque variable	15
Figure 5 : Proportion de chaque modalité de la variable grav avant et après traitement.....	16
Figure 6 : Liste des variables présentes après traitement du jeu de données	18
Figure 7 : Valeurs du V de Cramer pour chaque variable par ordre d'importance	19
Figure 8 : Regroupement des catégories pour la variable catv.....	20
Figure 9 : Répartition de la gravité en fonction de la catégorie de véhicule	21
Figure 10 : Valeurs du V_Cramer pour les modalités de la catégorie de véhicule initiales et recodées.....	21
Figure 11 : Numérotation des places dans la variable place.....	22
Figure 12 : (a). Comparaison du χ^2 et du V de Cramer pour les variables prox_pt_choc et choc, (b). Tableau de contingence de la variable prox_pt_choc avec la variable cible	23
Figure 13 : Répartition des modalités de la variable place	23
Figure 14 : χ^2 et V de Cramer pour les différentes modalités (a) de la variable place, (b). de la variable place_rec	24
Figure 15 : Corrélation entre les différentes modalités de place_rec et de la variable cible. Les niveaux de gravité 1, 2, 3 et 4 correspondent respectivement à indemne, tué, blessé hospitalisé et blessé léger.....	24
Figure 16 : Dichotomisation des équipements de sécurité	25
Figure 17 : χ^2 et V de Cramer pour les différentes variables équipement et catégorie de véhicule recodées	26
Figure 18 : Corrélation entre les différentes variables équipement, les différentes modalités de véhicule recodées et de gravité.....	26
Figure 19 : (a). Proportion de la gravité selon les conditions atmosphériques, (b). χ^2 et V de Cramer pour les différentes modalités de la variable atm	27
Figure 20 : (a). Corrélation entre la variable atm et les différentes modalités de la variable cible, (b). χ^2 et V de Cramer pour la variable atm recodée en binaire.	28
Figure 21 : Regroupement des catégories pour la variable manv	28
Figure 22 : Proportion de la gravité selon la manœuvre principale avant l'accident.....	29
Figure 23 : (a). Corrélation entre les différentes modalités de la variable manv et de la variable cible, (b). χ^2 et V de Cramer pour les différentes modalités de la variable manv	29
Figure 24 : Distribution de age_usager. (a). Histogramme des valeurs, (b). Graphique des quantiles, (c). Boîte à moustaches	30
Figure 25 : Distributions des variables lat et long	30
Figure 26 : Boîtes à moustaches des variables lat et long	31
Figure 27 : Distributions des variables lat et long, pour la métropole uniquement	31
Figure 28 : Boîtes à moustaches des variables lat et long pour la métropole uniquement...	31
Figure 29 : Proportion mensuelle de chaque classe de gravité	31
Figure 30 : Courbes du nombre d'usagers pour chaque classe de gravité selon le mois pour les années 2019 à 2022.....	32
Figure 31 : Proportion des différentes modalités de gravité selon s'il s'agit d'un jour de vacances et jours fériés ou non.....	33
Figure 32 : χ^2 et V de Cramer pour la variable jour_chome	33
Figure 33 : Courbes du nombre d'usagers pour chaque classe de gravité selon le jour de la semaine pour les années 2019 à 2022	33

Figure 34 : Tableaux de contingence pour la variable weekend (a) excluant le vendredi, (b) incluant le vendredi	34
Figure 35 : χ^2 et V de Cramer pour les variables (a). week-end sans le vendredi, (b). weekend avec le vendredi, (c) jour_semaine.....	34
Figure 36 : Répartition de la gravité pour la variable weekend	35
Figure 37 : Courbes du nombre d'usagers pour chaque classe de gravité selon l'heure pour les années 2019 à 2022.....	35
Figure 38 : Carte de la localisation des accidents selon la gravité en France métropolitaine	36
Figure 39 : Cartes (a) de la répartition des états de gravité des accidents par région (La taille des camemberts est proportionnelle au nombre d'usagers impliqués dans les accidents de chaque région), (b) des proportions de victimes décédées selon le département de France métropolitaine.	37
Figure 40 : Graphique de la répartition des modalités de la gravité en fonction de l'âge	37
Figure 41 : Graphique de la mortalité des accidents de la route selon le sexe. Le calcul des proportions se base sur 9556 hommes tués, contre 2645 femmes (soit une mortalité masculine 3 à 4 fois supérieure).	38
Figure 42 : Proportions, par classes d'âge, des catégories de véhicules associées aux usagers décédés	38
Figure 43 : Comparaison de la position de la personne accidentée selon le sexe	39
Figure 44 : Évolutions du nombre de cas par jour entre 2019 et 2022.....	42
Figure 45 : Lissage de la courbe sur une fenêtre glissante de 365 jours.	43
Figure 46 : Moyennes annuelles du nombre de cas par jours pour les années 2021 et 2022	44
Figure 47 : Décompositions des séries temporelles	44
Figure 48: Baselines avec shift de 1 jour.....	45
Figure 49 : Baselines moyennées sur 7 jours + shift de 1 jour	46
Figure 50 : Résultats de SARIMAX avec en variable exogène une série de Fourier	48
Figure 51 : Évaluations de SARIMAX avec en variable exogène une série de Fourier sur train et test.....	49
Figure 52 : Évaluation de MSTL avec différents trend_forecaster sur train et test	51
Figure 53 : Évaluation de PROPHET avec différentes saisonnalités sur train et test.....	56
Figure 54 : Évaluations de LSTM sur train et test.....	61
Figure 55 : Comparaison des modèles et prédictions à 1 et 6 mois pour les indemnes.....	63
Figure 56 : Comparaison des modèles et prédictions à 1 et 6 mois pour les blessés légers	64
Figure 57 : Comparaison des modèles et prédictions à 1 et 6 mois pour les blessés hospitalisés	65
Figure 58 : Comparaison des modèles et prédictions à 1 et 6 mois pour les tués	66
Figure 59 : Métriques (a) et matrice de confusion (b) du modèle de régression logistique de référence.....	67
Figure 60 : Résultats du GridSearchCV sur penalty, le type d'approche et C, hyperparamètres de la régression logistique	68
Figure 61 : Métriques (a) et matrice de confusion (b) du modèle de régression logistique optimisé	69
Figure 62 : Coefficients du modèle de régression logistique optimisé lors de la validation croisée	70
Figure 63 : Variance expliquée, en cumul (axe de gauche) ou unitairement (axe de droite), en fonction du nombre de composantes retenues.....	71
Figure 64 : Les 20 variables contribuant le plus aux deux premières composantes de l'analyse factorielle	71

Figure 65 : Métriques (a) et matrice de confusion (b) du modèle LinearSVC précédé d'une approximation de Nyström	73
Figure 66 : Variables sélectionnées selon le modèle avec leur importance	74
Figure 67 : Métriques et matrice de confusion selon le modèle	75
Figure 68 : Métriques et matrice de confusion avec les paramètres optimisés selon le modèle	77
Figure 69 : Métriques et matrice de confusion pour le meilleur modèle	78
Figure 70 : Importance des variables pour le meilleur modèle Random Forest	78
Figure 71 : Graphiques d'importance des variables selon SHAP	79
Figure 72 : Graphiques de densité des valeurs de SHAP	80
Figure 73 : Graphique à coordonnées parallèles présentant le lien entre les hyperparamètres de Catboost et la performance du modèle.....	82
Figure 74 : Métriques (a) et matrice de confusion (b) du modèle CatBoost	83
Figure 75 : Importance des variables pour le meilleur modèle CatBoost	83
Figure 76 : Graphiques de densité des valeurs de SHAP pour le modèle optimisé CatBoost	84
Figure 77 : Courbes de mlogloss et mrror suivant le nombre d'arbres.....	86
Figure 78 : Matrices de confusion pour chaque méthode	86
Figure 79 : Rapports de classification selon la méthode.....	87
Figure 80 : Les 20 premières features importance selon weight, gain, cover et selon la méthode.....	88
Figure 81 : Courbes de mlogloss et mrror pour le modèle optimisé.....	89
Figure 82 : Matrices de confusion du modèle de référence et du modèle optimisé.....	89
Figure 83 : Rapports de classification pour le modèle de référence et le modèle optimisé ..	90
Figure 84 : Graphiques d'importance des variables selon SHAP	91
Figure 85 : Graphiques de densité des valeurs de SHAP.....	92
Figure 86 : Métriques et matrice de confusion selon le jeu de données binaire utilisé	98
Figure 87 : Réarrangement de la matrice de confusion pour de la classification multilabels et comparaison avec la classification binaire.....	101
Figure 88 : Importance des variables selon le jeu de données binaires.....	102
Figure 89 : Graphiques d'importance des variables selon SHAP selon le jeu de données binaires	103
Figure 90 : Graphique de densité des valeurs de SHAP selon le jeu de données binaires	104
Figure 91 : Comparaison des métriques et matrice de confusion pour le jeu de données initial et le jeu de données avec modification de variables	106
Figure 92 : Diagrammes radars de comparaison des modèles en termes de Precision, Recall, Specificity, F1-Score et Index Balanced Accuracy	107
Figure 93 : Métriques et matrice de confusion pour le meilleur modèle	108
Figure 94. Risque relatif (échelle log) d'être hospitalisé ou tué pour une personne de la base de données	109
Figure 95 : Courbes de perte et d'accuracy en fonction du nombre d'époques pour le modèle de référence Keras	111
Figure 96 : Métriques et matrice de confusion pour le modèle de référence (Keras)	111
Figure 97 : Résultats des modèles Keras avec ré-équilibrage des classes	112
Figure 98 : Résultats des modèles de Deep Learning (Keras) optimisés.....	114
Figure 99 : Courbes de perte et d'accuracy selon le nombre d'époques pour le modèle de référence (PyTorch)	115
Figure 100 : Métriques et matrice de confusion pour le modèle Deep Learning de référence (PyTorch)	115

Figure 101 : Comparaison des performances en fonction (a) du taux de dropout choisi en sortie de couche Dense, (b) du nombre maximal de neurones dans le modèle.....	116
Figure 102 : Parallel Coordinate Plot pour l'hyperparamétrage du modèle TabNet avec Optuna	117
Figure 103 : Métriques et matrice de confusion pour le modèle TabNet optimisé	117
Figure 104 : (a) Importance des variables dans le modèle TabNet (b). Masque d'activation des variables pour les 50 premières observations de l'échantillon de test.	118
Figure 105 : Histogramme du nombre d'usagers impliqués dans les accidents, et fonction de répartition de cette variable.	119
Figure 106 : Comparaison des résultats des modèles Random Forest, avec ou sans la variable nb_usagers_gr.....	120
Figure 107 : Comparaison des performances des modèles de Machine Learning et de Deep Learning, pour chaque état de gravité.....	121
Figure 108 : Ordre d'importance des variables explicatives dans les modèles	122

INTRODUCTION AU PROJET

I.1 Contexte

La France dispose d'infrastructures routières particulièrement importantes avec environ 1,7 millions de kilomètres de routes en 2021 déclinées comme ceci (Routes de France, 2023) :

Longueur du réseau routier français métropolitain [2021]

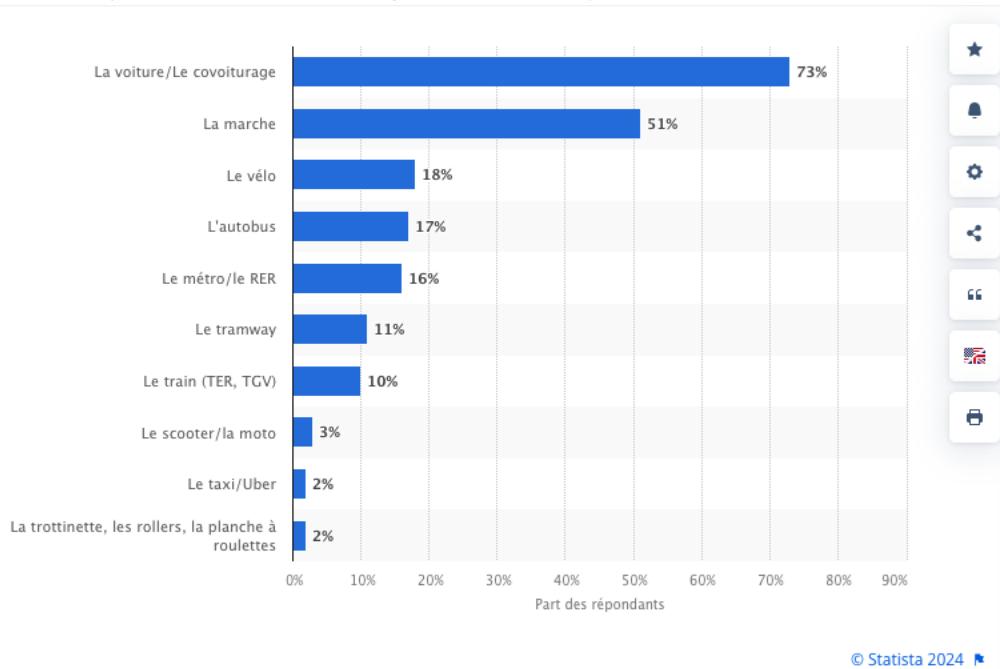
	Km	% du trafic	Observations
Autoroutes concédées	9 221	16 %	dont 2 372 Km à 2 x 3 voies
Autoroutes non concédées	3 309	16 %	
Routes nationales	8 380	4 %	dont environ 2 836 km à chaussées séparées
Routes départementales	378 834	64 %	dont environ 1 500 km à chaussées séparées
Routes communales et rues	705 000		
Total	1 104 744		
Chemins ruraux	env. 600 000 km		

Sources : Cerema, ASFA, SDES.

Au 1er janvier 2023, sur ces routes circulait le parc automobile français suivant (Ministère de la transition écologique et de la cohésion des territoires, 2024) :

- 38,9 millions de voitures particulières,
- 6,4 millions de véhicules utilitaires légers (VUL),
- 620 000 poids lourds,
- 94 000 autobus et autocars.

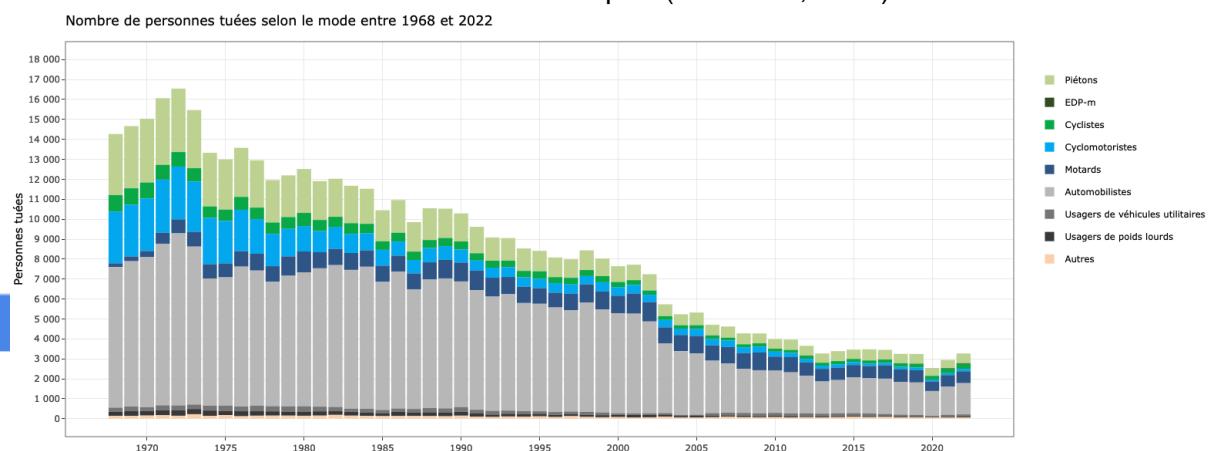
Les Français n'utilisent pas uniquement ces types de véhicules pour leurs déplacements. En effet, les moyens de transport utilisés par les Français pour les déplacements quotidiens en 2023 sont (Statista Research Department, 2024) :



Malheureusement, chaque année, de nombreux accidents de la route se produisent. D'ailleurs, le bilan 2023 provisoire de la sécurité routière est le suivant (Observatoire national interministériel de la sécurité routière, 2024) :

- **3 402 personnes sont décédées** en 2023 sur les routes de France métropolitaine ou d'outre-mer (estimation ONISR au 22/01/2024),
- **232 000 personnes ont été blessées** en 2023 sur les routes de France métropolitaine, dont **16 000 gravement**, d'après la méthode d'estimation ONISR-Université Gustave Eiffel (Registre du Rhône).

En regardant plus précisément le nombre de personnes tuées selon le mode de transport utilisé, on voit une évolution du nombre de morts qui a diminué depuis 1968 pour stagner depuis 2013 (hors année 2020 impactée par le covid). De plus, on se rend compte que la mortalité est différente selon le mode de transport (CEREMA, 2024):



I.2 Objectifs

Le projet a pour but de **prédirer la gravité des accidents routiers en France** d'après des données historiques.

Plus spécifiquement, l'objectif ici sera de **prédirer la catégorie de gravité de l'accident** (indemne, blessé léger, blessé hospitalisé, décès) pour chaque usager entré dans la base de données, en fonction de caractéristiques individuelles (âge, sexe, utilisation d'équipements de sécurité, place dans le véhicule...), des caractéristiques de son mode de transport (voiture, 2-roues motorisé, vélo, transport en commun...), des caractéristiques du lieu de l'accident (type de voie, intersection, double-sens...), et de caractéristiques contextuelles (date, heure, luminosité, météo...).

Dans un deuxième temps, **le poids de chaque facteur dans la classification sera étudié**, afin de classer les facteurs par ordre d'importance : qu'est-ce qui fait qu'une personne va être plus sévèrement atteinte lors d'un accident routier ? Cette étape *d'interprétabilité* du modèle permettra de dégager les principaux axes d'amélioration de la sécurité routière afin de réduire l'accidentalité à différents niveaux :

- Matériel : équipement de sécurité, type de véhicule...
- Humain : genre, âge, mode de déplacement...
- Localisation : type de route, intersection, urbain ou non...
- Conditions atmosphériques : météo, luminosité...

Une troisième partie, si le temps le permet, pourrait être de **simuler, à partir du modèle, la gravité des accidents si l'on améliorait certains des facteurs identifiés** : par exemple, s'il ressort que les équipements de sécurité jouent un rôle important, de combien aurions-nous pu réduire le nombre de morts en 2022 si toutes les voitures étaient équipées d'airbag ?

L'étude de ce projet est réalisée par l'équipe suivante :

- **Camille Pelat** : statisticienne chez Santé publique France, j'ai plus souvent affaire à des outcomes binaires (malade / non malade), parfois multinomiaux (allaitement exclusif / mixte / non), et plus souvent dans une visée explicative que prédictive. Le rééquilibrage du jeu de données, la sélection des variables par cross-validation et le test de performance sur un jeu de test sont des étapes que je trouve intéressantes à aborder dans ce projet. L'étape d'interprétabilité me paraît aussi importante pour faire le lien entre prédiction et explication.
- **Matthieu Claudel** : ingénieur d'études en maintenance chez SNCF Matériel, j'ai participé dans mon précédent poste à un projet de prédiction basé sur le computer vision mais jamais sur des données numériques ou catégorielles. Ce projet représente pour moi une opportunité de compléter de nouvelles compétences.
- **Nadège Reboul** : enseignant-chercheur dans le domaine du génie civil, actuellement en disponibilité pour reconversion, j'ai travaillé par le passé sur des données quantitatives, issues de mesures expérimentales, et utilisé des méthodes de clustering sous R et/ou matlab. Le travail avec Python, sur des données fortement catégorielles, avec une finalité de classification supervisée est donc une première expérience pour moi.
- **Nicetas Tevoedjre** :
- **Vanessa Ibert** : docteur en chimie organique et professeur des écoles en reconversion professionnelle. En dehors de la formation, je n'ai pas encore eu l'occasion de traiter de telles problématiques.

Des recherches dans la littérature montrent des projets similaires. Nous avons sélectionné trois d'entre eux :

- La première étude est un article paru dans la revue de l'IA (Talbi, 2020). Dans cet article, le nettoyage du jeu de données est très rapide (suppression de colonnes) sans faire de regroupement de modalités (hormis les variables 'lat' et 'long'). Il est à noter le travail de regroupement des latitudes et des longitudes en 15 modalités en utilisant la méthode des K-Means. Pour la modélisation, il compare les algorithmes de Random Forest et de XGBoost avec de meilleurs résultats avec XGBoost. En perspectives d'améliorations, il propose l'optimisation des hyperparamètres et le rééquilibrage des catégories de la variable cible.
- La seconde proposition est un repo GitHub (Maxime, 2019). Ce repos traite plus en profondeur le nettoyage des données : suppression de colonnes et remplacement des valeurs manquantes (généralement par la modalité la plus fréquente). En revanche, la variable cible est remaniée pour obtenir une variable cible binaire (regroupement des modalités 1 et 4 pour devenir 'Light Injury', regroupement des modalités 2 et 3 pour devenir 'Serious Injury and Death'). Pour la modélisation, l'algorithme Random Forest est utilisé dans un premier temps. Puis les variables les plus importantes sont sélectionnées pour refaire un entraînement avec uniquement ces variables. Enfin, une optimisation des hyperparamètres est effectuée.

- Le troisième est un article néo-zélandais, **Source spécifiée non valide.**, dont les données et l'objectif sont très proches des nôtres. Leur objectif est de prédire la gravité d'un accident (et pas d'un usager accidenté, à la différence de notre projet) en 4 classes : accident sans blessé, avec blessé léger, avec blessé sévère, avec tué. Leur analyse contient une étape de rééquilibrage du jeu de données, des méthodes d'ensemble (XGBoost notamment), une étape d'interprétabilité et de sélection des variables les plus contributives, et un ré-entraînement du modèle sur ce sous-ensemble de variables.

I COMPREHENSION ET MANIPULATION DES DONNEES

I.1 Cadre

Afin de réaliser le projet, nous avions la possibilité d'utiliser des jeux de données publiques des deux sites suivants :

- [Bases de données annuelles des accidents corporels de la circulation routière - Années de 2005 à 2021 - data.gouv.fr](https://www.data.gouv.fr/datasets/bases-de donnees-annuelles-des-accidents-corporels-de-la-circulation-routiere-annnees-de-2005-a-2021/) (Ministère de l'Intérieur et des Outre-Mer, 2013, māj 2023)
- <https://www.kaggle.com/ahmedlahlou/accidents-in-france-from-2005-to-2016> (Lahlou Mimi, 2018)

Après étude des données du site Kaggle, nous nous sommes rendu compte qu'elles étaient issues du site du gouvernement et ne concernaient que les années 2005 à 2016. Nous avons donc opté pour **l'utilisation des données gouvernementales**. Cette base de données recense **l'ensemble des accidents corporels survenus en France entre 2005 et 2022**.

Est considéré comme accident corporel, « **tout accident survenu sur une voie ouverte à la circulation publique, impliquant au moins un véhicule et ayant fait au moins une victime ayant nécessité des soins** ». Lorsqu'un tel accident survient, les forces de l'ordre interviennent sur place et remplissent des Bulletins d'Analyse des Accidents Corporels (BAAC) administrés par l'Observatoire National Interministériel de la Sécurité Routière (ONISR). Ce sont les éléments recensés dans ces BAAC, et donc **potentiellement sujets à des erreurs de saisie ou des défauts de saisie** (les forces de l'ordre ne sont pas toujours informées lorsque l'accident n'est pas mortel), qui constituent notre base de données.

Certaines informations qui pourraient nuire à la vie privée des usagers concernés (conduite sous l'emprise d'alcool ou de drogue, défaut de permis de conduire...) **ont été éliminées** avant la diffusion de cette base de données au grand public. **Il est probable que ces différents facteurs exercent une influence non négligeable sur l'accidentalité routière et cela devra être conservé à l'esprit à la lecture des conclusions du présent rapport.**

Les bases de données sont annuelles et se composent chaque année de 4 fichiers au format .csv : «Caractéristiques - Lieux - Véhicules - Usagers ». La Figure 1 présente l'ensemble des variables contenues dans ces fichiers. Il est intéressant de noter que les fichiers Caractéristiques, Lieux et Véhicules pourront être fusionnés grâce à l'identifiant de l'accident (Num_Acc), tandis que le fichier Usagers pourra être relié aux autres par l'intermédiaire du fichier Véhicules puisqu'ils ont en commun les identifiants du véhicule (id_vehicule et num_veh). **Pour chaque usager, il est donc possible d'avoir accès à l'ensemble de ces variables.**

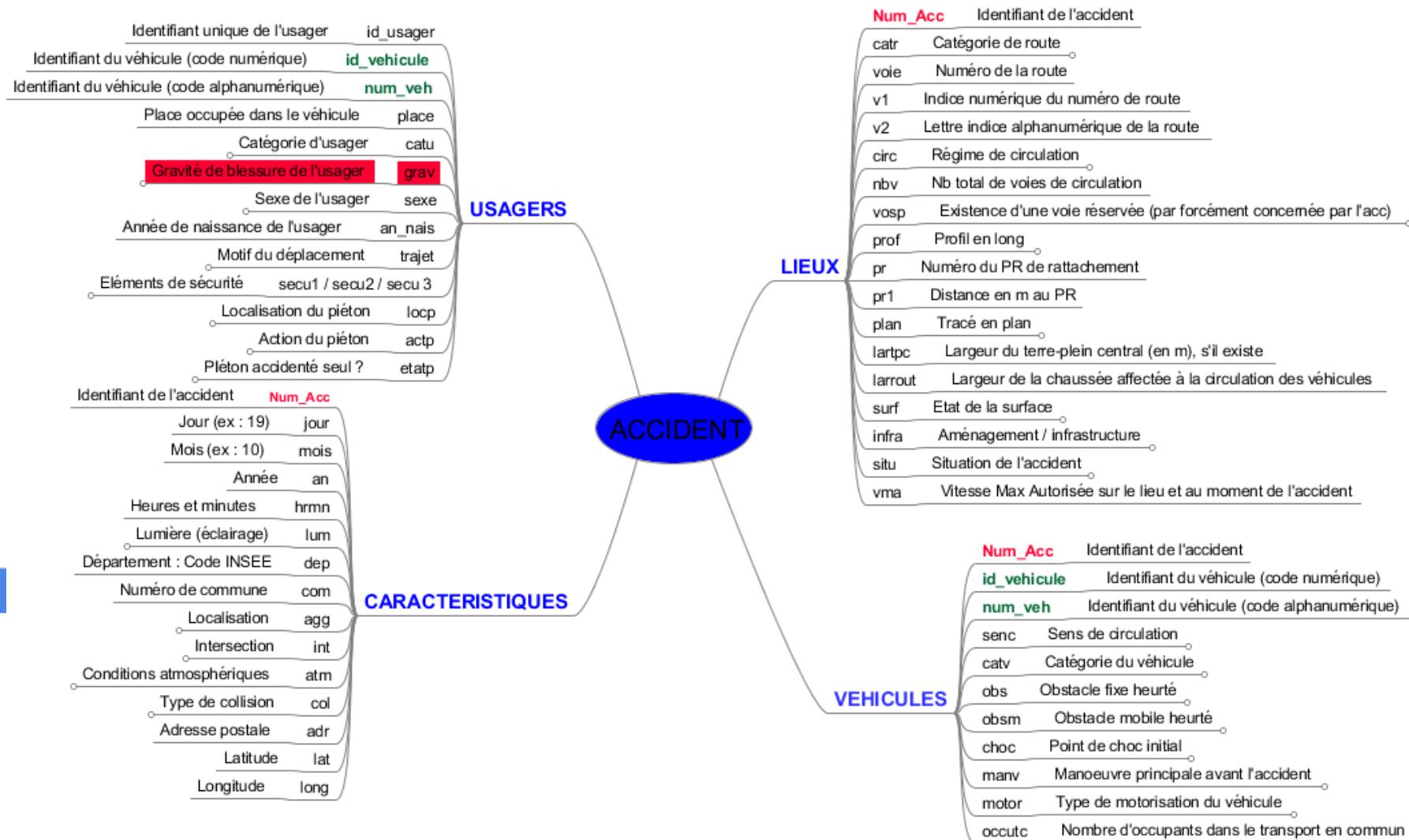


Figure 1 : Liste des variables présentes dans les bases de données mises à disposition du grand public

La fusion de l'ensemble des fichiers crée un jeu de données de plus de 2 millions de lignes. Une première étude du nombre d'accidents et de tués par an (Figure 2) montre une diminution du nombre de 2005 à 2013, puis une stabilisation du nombre d'accidents à environ 130000 par an et du nombre de tués à environ 3500 par an (en dehors de l'année 2020 correspondant à l'année du confinement).

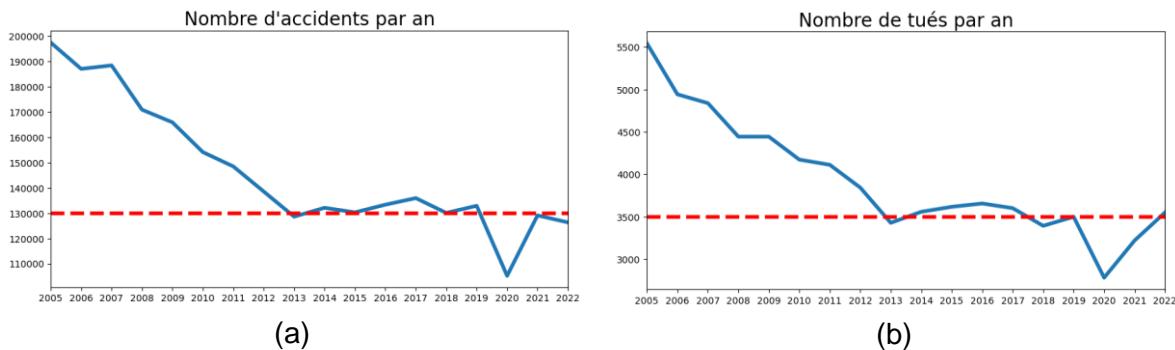


Figure 2 (a). Nombre d'accidents par an, (b). Nombre de tués par an

A la suite de modifications du processus de saisie des forces de l'ordre, les données depuis l'année 2018 ne peuvent être comparées à celles des années précédentes. Et, depuis 2019, des territoires d'Outre-mer ont été ajoutés, de nouvelles variables ont été créées (vma, motor, secu1, secu2, secu3 et id_vehicules), alors que d'autres ont été supprimées (gps et secu). En conséquence, nous avons fait le choix de **réduire notre analyse sur la période 2019 à 2022**. La fusion de l'ensemble des fichiers sur cette période permet d'obtenir **un jeu de données de 494 182 lignes avec 55 variables**.

I.2 Pertinence

Parmi ces variables, compte-tenu de la problématique de notre projet, la **gravité de blessure de l'usager (grav)** retient particulièrement notre attention. Chaque usager peut être considéré comme :

- tué s'il décède du fait de l'accident, sur le coup, ou dans les 30 jours qui suivent l'accident,
- blessé hospitalisé, s'il est hospitalisé plus de 24 heures,
- blessé léger, s'il a reçu des soins médicaux mais n'a pas été admis à l'hôpital plus de 24 heures,
- ou indemne.

Cette variable, grav, sera la variable cible de notre problème de prévision. Les 54 autres variables seront des variables explicatives.

Afin de vérifier si le jeu de données n'est pas biaisé (surtout sur l'année 2020 à cause du confinement), nous vérifions si le pourcentage d'accidents selon la gravité est homogène chaque année (Figure 3).

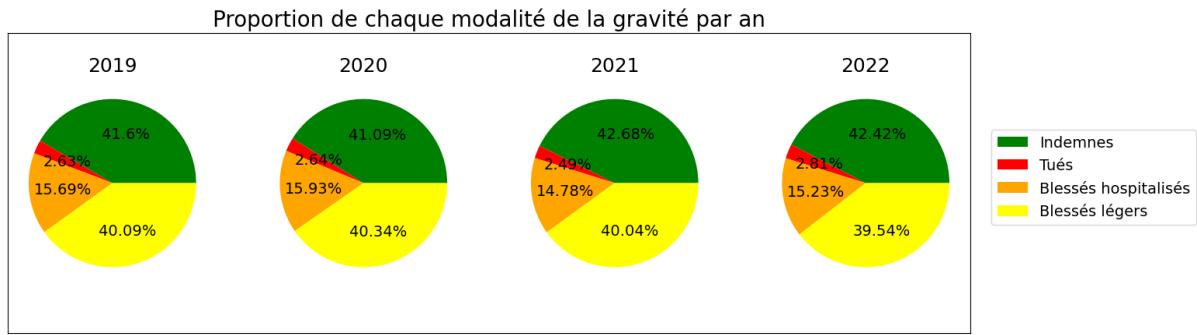


Figure 3 : Pourcentage d'accidents selon la modalité pour les années 2019 à 2022

Nous pouvons voir qu'il y a très peu de différence de proportion pour chaque modalité quelle que soit l'année. Donc nous pouvons garder le jeu de données sur les années 2019 à 2022 pour essayer de prédire la gravité des accidents routiers en France.

I.3 Pre-processing et feature engineering

Démarche globale

Le jeu de données résultant de la fusion des fichiers .csv usagers, véhicules, lieux et caractéristiques 2019 à 2022 contenait 55 variables dont une variable cible, 4 variables "identifiants" ('Num_Acc', 'id_vehicule', 'num_veh','id_usager') et 50 variables potentiellement explicatives.

Nous avons d'abord supprimé **164 doublons** (lignes exactement identiques).

Puis nous avons regardé le pourcentage de valeurs manquantes dans chaque variable. Parmi nos 55 variables, **14 avaient plus de 8% de valeurs manquantes** (Figure 4). Il s'agissait majoritairement de :

- variables "**administratives**" (ex. "v1 : Indice numérique du numéro de route, "id_usager : Identifiant unique de l'usager", "pr1 : Distance en mètres au PR (par rapport à la borne amont)."),
- variables renseignées uniquement dans un **sous-ensemble des accidents** (ex : "lartpc : Largeur du terre-plein central (TPC) s'il existe", "occutc : Nombre d'occupants dans le transport en commun", "locp : Localisation du piéton :'), donc à faible potentiel explicatif.

Nous les avons donc toutes supprimées, à l'exception des variables secu3 (98% de valeurs manquantes) et secu2 (39% de valeurs manquantes).

Les variables secu2 et secu3 sont en effet deux variables indiquant l'utilisation d'équipements de sécurité, complémentaires dans leur construction à secu1, et qui feront l'objet d'un recodage spécial, qui sera abordé dans la section "exemples détaillés avec visualisation".

Nous supprimons aussi les 4 variables "identifiants" ayant servi à faire la jonction entre les différents fichiers csv.

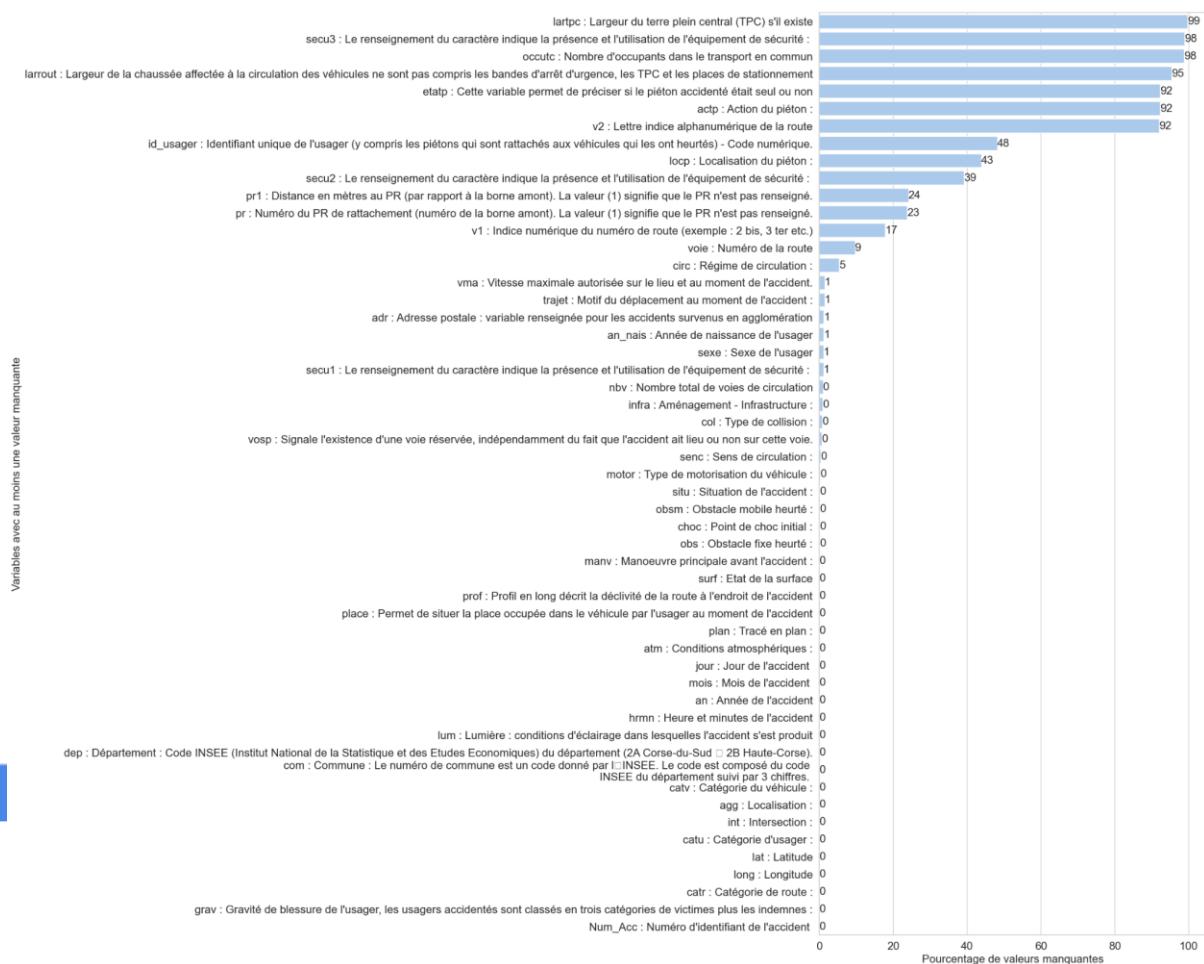


Figure 4 : Pourcentage de valeurs manquantes pour chaque variable

Nous avons choisi de supprimer aussi :

- d'autres variables "**administratives**" ("adr : adresse postale", "senc : sens de circulation")
- d'autres variables renseignées uniquement dans un **sous-ensemble des accidents** ("vosp : présence d'une voie réservée, indépendamment du fait que l'accident ait eu lieu ou non sur cette voie"),
- des variables **trop corrélées** entre elles ("com : numéro de commune" corrélée avec "dep : numéro de département", "nbv : nombre total de voie de circulation" et "vma : vitesse maximale autorisée" corrélées avec "catr : catégorie de route", "catu : catégorie d'usagers" corrélée avec "place" que l'on va modifier),
- des variables possédant trop de valeurs manquantes et ne pouvant pas être renseignée sans risquer de biaiser le jeu de données ("trajet : motif de déplacement au moment de l'accident").

Étant donné la taille de notre jeu de données et compte-tenu de la faible proportion de valeurs manquantes pour les autres variables, nous avons décidé de supprimer les lignes avec des valeurs manquantes pour les autres variables. In fine, le retrait de ces valeurs manquantes a majoritairement impacté la catégorie des personnes indemnes (11.0 %), puis celle des blessés légers (8.7 %), puis celle des blessés hospitalisés (7.9 %), et enfin celle des tués (6.5 %). Ce choix n'a donc pas eu de répercussion sur l'une des modalités de la variable cible en particulier et n'a pas aggravé le déséquilibre du jeu de données initial.

Nous avons aussi modifié les modalités de certaines variables. (cf. tableau résumé des traitements de chaque variable).

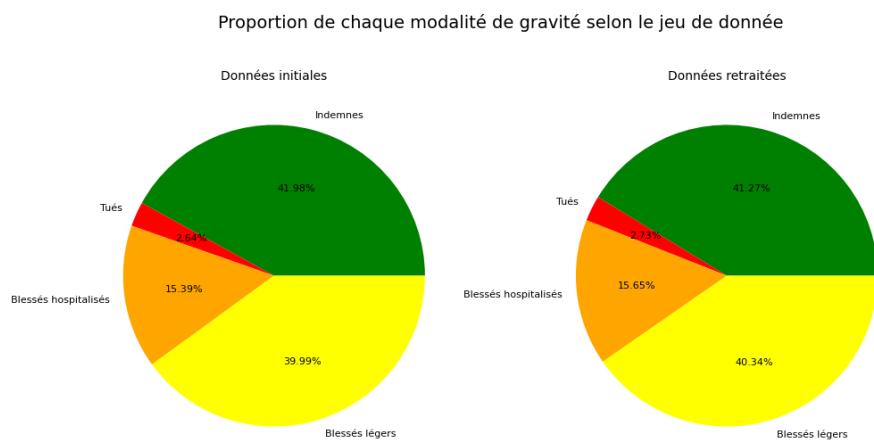
Enfin, nous avons créé des variables qui nous paraissent plus pertinentes.

Après ces traitements, nous obtenons un jeu de données avec **447 136 lignes et 41 variables**. Certaines de ces variables, comme 'dep' et 'an' ne sont conservées que pour la data visualisation et seront supprimées pour la modélisation.

Enfin, nous réalisons un traitement spécifique pour la modélisation :

- création de dummies pour les variables catégorielles non binaires (avec conservation de tous les dummies dans le but de faire un Random Forest et /ou un XGBoost). Dans le cas où nous utiliserions d'autres algorithmes sensibles à la multi-colinéarité, nous enlèverions une colonne de chaque variable dummisée.
- réflexion sur les modes de normalisation/standardisation des variables continues (age_usager, latitude, longitude, mois et heure)

Finalement, le jeu de données final est composé de **447136 lignes et 98 variables**. Le traitement des données n'a pas changé les proportions des modalités de la variable cible (Figure 5). Cependant, le jeu de données reste largement **déséquilibré**. Des procédures de **ré-échantillonnage** devront donc être mises en œuvre au moment de la modélisation.



L'ensemble des modifications effectuées est regroupé dans le tableau de synthèse (Tableau 1), les variables de la base de données finalisée sont présentées sur la Figure 6 et la valeur du V_Cramer pour chaque variable est présentée par ordre d'importance dans la Figure 7.

Tableau 1 : Synthèse des traitements effectués sur les variables

Variables	Nombre NaN	Gestion NaN	Gestion valeurs aberrantes	Catégorisation	Traitement pour modélisation
Num_Acc	0	Suppression de la colonne	X	X	X
jour	0	X	X	Création variable 'Weekend': 0 - non 1 - oui Création variable 'jour_chome': 0 - non 1 - oui X	X
mois	0	X	X	X	X
an	0	X	X	X	Supprimé pour la modélisation
hmn	0	X	X	Création colonne heure et suppression 'hmn' Regrouper les catégories 3 et 4 pour devenir : 0 - Plein jour (1) 1 - Crémouille ou aube (2) 2 - Nuit sans éclairage (3 et 4) 3 - Nuit avec éclairage public (5)	X
lum	9	Suppression des lignes avec NaN	X	Dummies	
dep	0	X	X	Remplacement des '1'... '9' par '01'... '09'	Supprimée pour la modélisation
com	0	Suppression de la colonne	X	X	X
agg	0	X	X	Remplacement de 1 en 0 et de 2 en 1 (pour être binaire)	X
int	21	Suppression des lignes avec NaN	X	Regrouper les catégories en 2 catégories: 0 - hors intersection (1) 1 - hors intersection (2 à 9)	X
atm	33	Suppression des lignes avec NaN	X	Regrouper les catégories en 2 catégories: 0 - Normal (1) 1 - Autres (2, 3, 4, 5, 6, 7, 8, 9)	X
col	3870	Suppression des lignes avec NaN	X	X	Dummies
adr	6042	Suppression de la colonne	X	X	X
lat	0	X	Beaucoup de valeurs aberrantes retraitées au cas par cas	X	X
long	0	X	Beaucoup de valeurs aberrantes retraitées au cas par cas	X	X
catr	0	X	X	X	X
voie	47519	Suppression de la colonne	X	X	X
v1	87996	Suppression de la colonne	X	X	X
v2	454208	Suppression de la colonne	X	X	X
circ	26227	Suppression des lignes avec NaN	X	Regrouper les catégories en 2 catégories: 0 - Unidirectionnel (1, 3, 4) 1 - Bidirectionnel (2)	X
nbv	4775	Suppression de la colonne	X	X	X
vosp	2830	Suppression de la colonne	X	X	X
prof	78	Suppression des lignes avec NaN	X	Regrouper en 2 catégories: 0 - plat (1) 1 - pentu (2, 3, 4)	X
pr	117212	Suppression de la colonne	X	X	X
pr1	118945	Suppression de la colonne	X	X	X
plan	60	Suppression des lignes avec NaN	X	Regrouper les catégories en 2 catégories: 0 - Rectiligne 1 - Courbe	X
lartpc	492635	Suppression de la colonne	X	X	X
larout	479093	Suppression de la colonne	X	X	X
surf	114	Suppression des lignes avec NaN	X	X	Dummies
infra	4458	Suppression des lignes avec NaN	X	X	Dummies
situ	274	Suppression des lignes avec NaN	X	X	Dummies
vma	7141	Suppression de la colonne	X	X	X
id_vehicule	0	Suppression de la colonne	X	X	X
num_veh	0	Suppression de la colonne	X	X	X
place	3	Suppression des lignes avec NaN	X	Création de la variable 'place_rec' regroupée en 4 catégories: 1 - Conducteur 2 - Passager avant 3 - Passager arrière 4 - Piéton	Dummies
catu	0	Suppression de la colonne	X	X	X
grav	0	Suppression des lignes avec NaN	X	X	X
sexe	5506	Suppression des lignes avec NaN	X	Remplacement de 1 par 0 et de 2 par 1 (pour être binaire)	X
an_nais	5641	Suppression des lignes avec NaN	Suppression des lignes où l'usager est une femme de 118 ans et + Suppression des lignes où l'usager est un homme de 112 ans et +	Création de la variable 'age_usager'	X
trajet	6600	Suppression de la colonne	X	X	X
secu1	5320			Créer 7 variables avec les modalités 0 (non) ou 1 (oui): 'Eq_ceinture' 'Eq_casque' 'Eq_dispositif_enfants' 'Eq_Gilet_refléchissant' 'Eq_Airbag' 'Eq_Gants' 'Eq_autre'	
secu2	193102	On remplace les Nan par des 0.	X		X
secu3	488124				
locp	216447	Suppression de la colonne	X	X	X
actp	455749	Suppression de la colonne	X	X	X
statp	455834	Suppression de la colonne	X	X	X
id_usager	238108	Suppression de la colonne	X	X	X
senc	1642	Suppression de la colonne	X	X	X
catv	13	Suppression des lignes avec NaN	X	Regrouper les catégories en 5 catégories: 0 - Véture (3, 7, 10) 1 - Moto (2, 30, 31, 32, 33, 34, 35, 36, 41, 42, 43) 2 - Poids lourds (13, 14, 15, 16, 17, 20, 21) 3 - Transport en commun (37, 38, 39, 40) 4 - Vélo/Trottinette (1, 50, 60, 80) 5 - Autre véhicule (0, 99)	Dummies
obs	163	Suppression des lignes avec NaN	X	Regrouper les catégories en 2 catégories: 0 - Sans obstacle 1 - Avec obstacle	X
obsm	231	Suppression des lignes avec NaN	X	Regrouper les catégories en 4 catégories: 0 - Siège conducteur 1 - Piéton 2 - Véhicule 3 - Animaux/Autres	Dummies
choc	208	Suppression des lignes avec NaN	X	Création de la variable 'proximite_choc': 0 - Pas de proximité 1 - Proximité	X
manv	148	Suppression des lignes avec NaN	X	Regrouper les catégories en 4 catégories: 0 - Même sens (1, 2, 3, 7, 25) 1 - Contre sens (4, 5, 8) 2 - Immobile (22, 23, 24) 3 - Changement de direction (6, 9, 10 à 21, 26)	Dummies
motor	977	Suppression des lignes avec NaN	X	X	Dummies
occute	487581	Suppression de la colonne	X	X	X

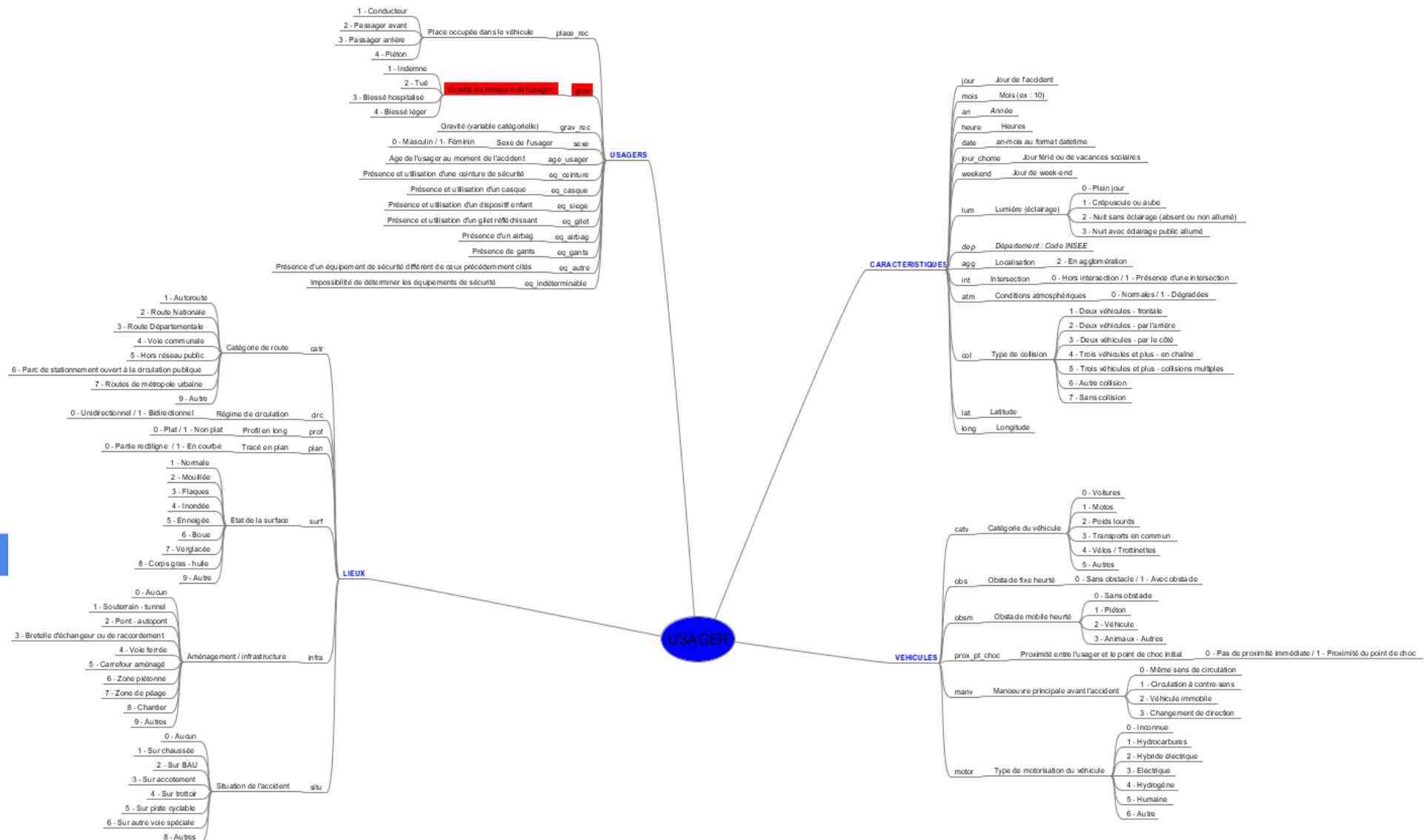


Figure 6 : Liste des variables présentes après traitement du jeu de données

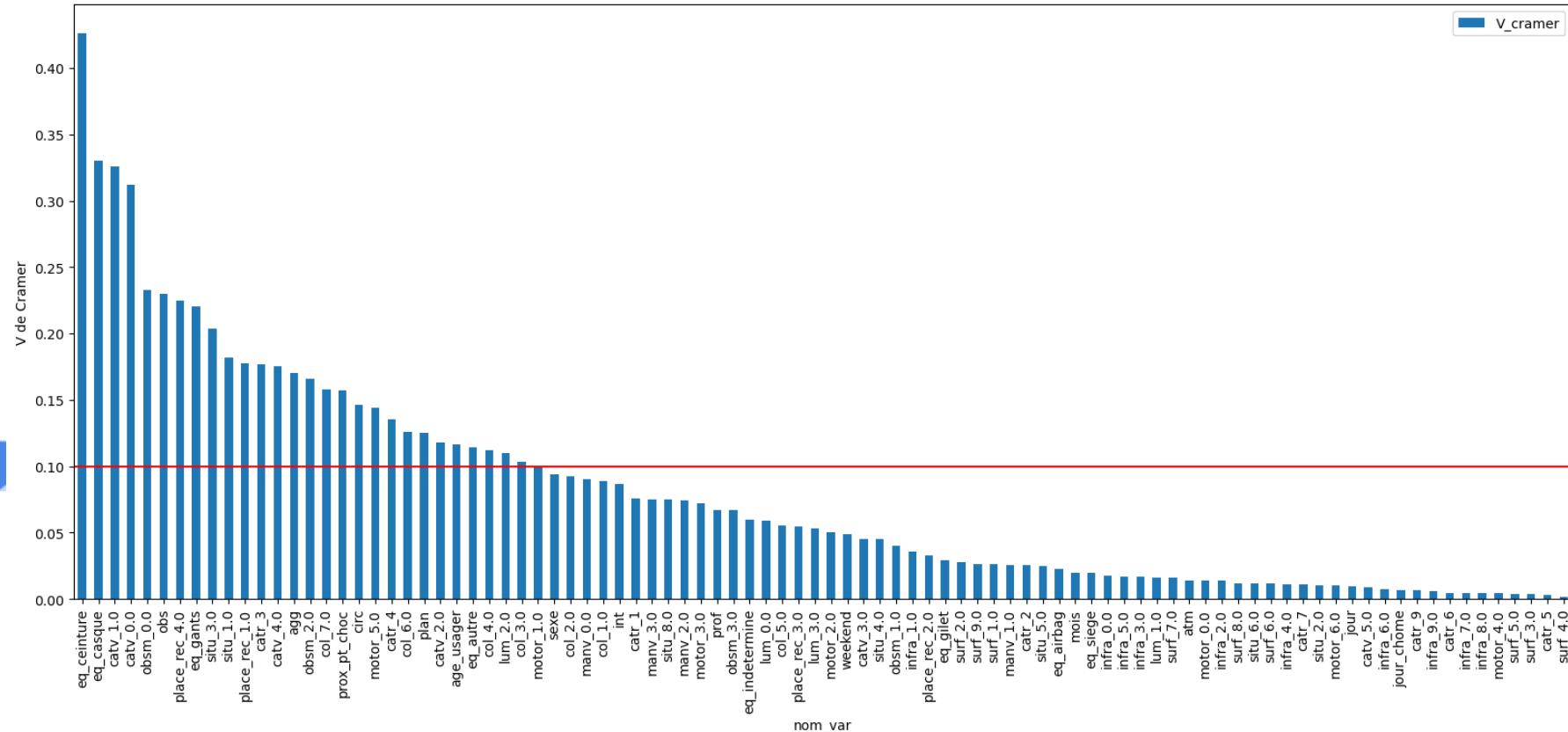


Figure 7 : Valeurs du V de Cramer pour chaque variable par ordre d'importance

I.4 Exemples détaillés avec visualisation

Nous présentons ci-après les argumentaires nous ayant conduit aux modifications de la base de données initiale pour 5 variables particulières.

CATÉGORIES DE VÉHICULES

Treatment de la variable catv

Il a donc été décidé de procéder à des regroupements, tels que présentés sur la Figure 8. La variable catv réencodée présente donc 6 modalités : 0 – Voiture, 1 – Moto, 2 – Poids lourds, 3 – Transport en commun, 4 – Vélo/Trottinette, 5 – Autre véhicule.

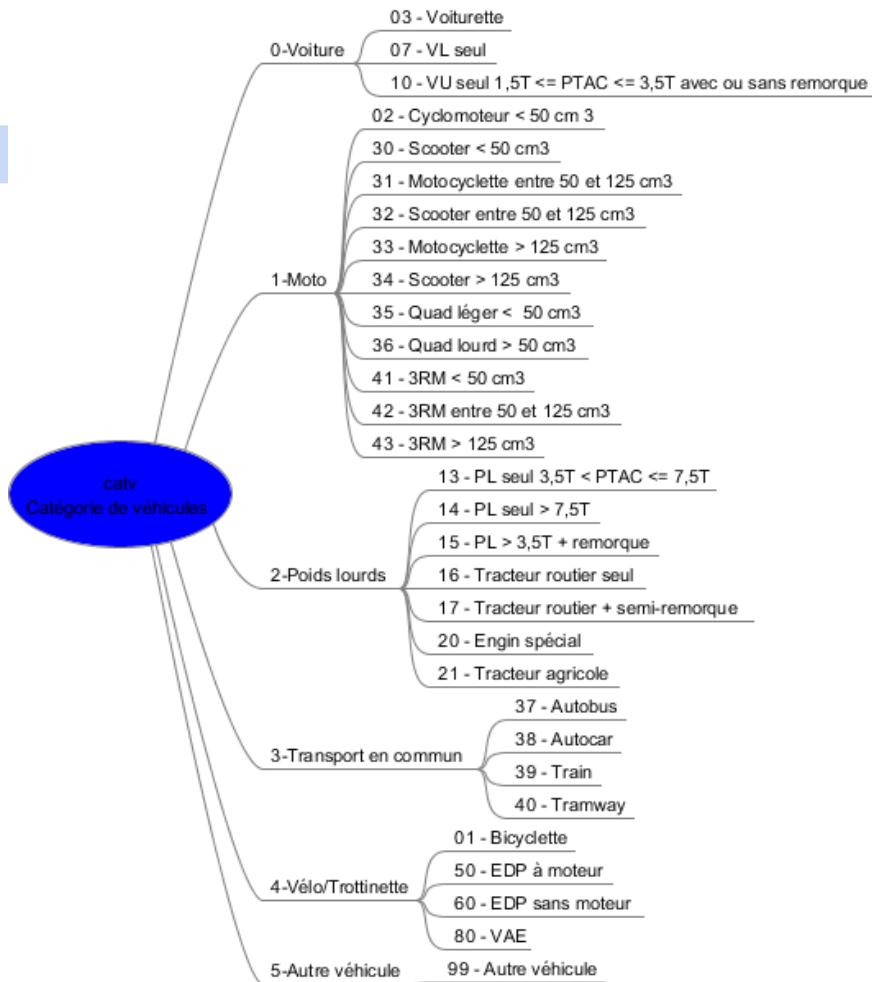


Figure 8 : Regroupement des catégories pour la variable catv

Le recodage en 6 modalités permet de conserver la différence de gravité selon le type de véhicule (Figure 9). On voit très nettement que la proportion de gravité des accidents est plus importante lorsque l'usager est en moto ou en vélo/trottinette. A l'inverse, les poids lourds sont ceux qui ont la proportion de personnes indemnes à l'issue d'un accident la plus élevée.

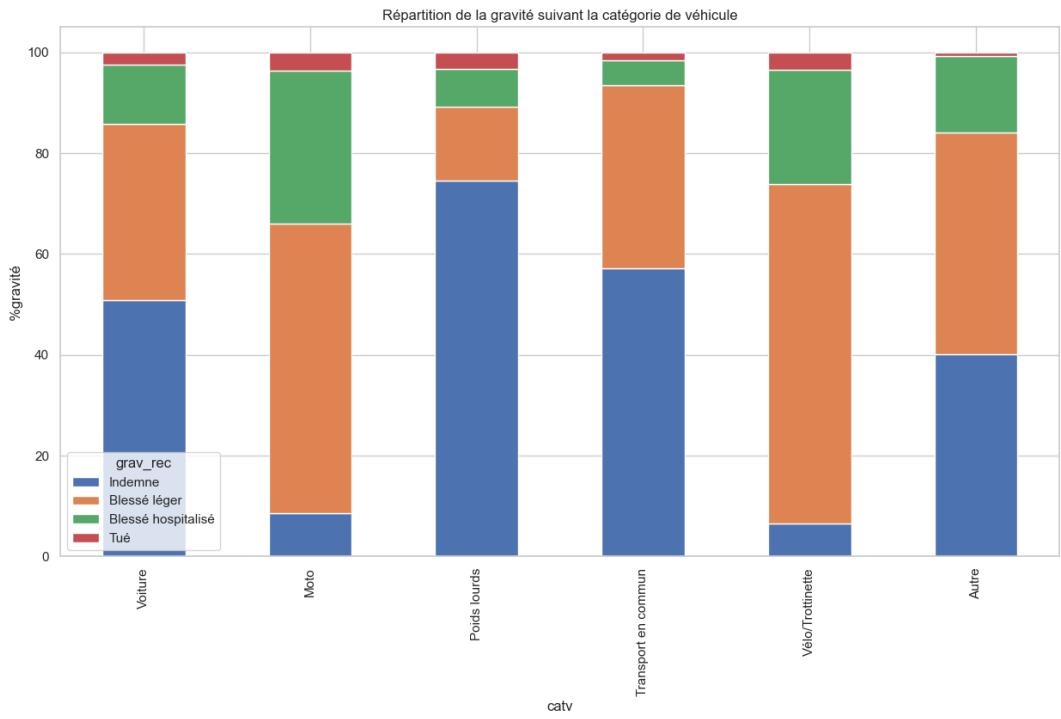


Figure 9 : Répartition de la gravité en fonction de la catégorie de véhicule

La Figure 10 présente les valeurs du V de Cramer pour les modalités de la nouvelle variable en rouge et de l'ancienne variable en bleu. Le nouvel encodage a tendance à réduire le nombre de modalités faiblement influentes et conserve les variables les plus influentes. Le regroupement a tendance à accentuer l'intensité de la relation des variables encodées avec la variable cible.

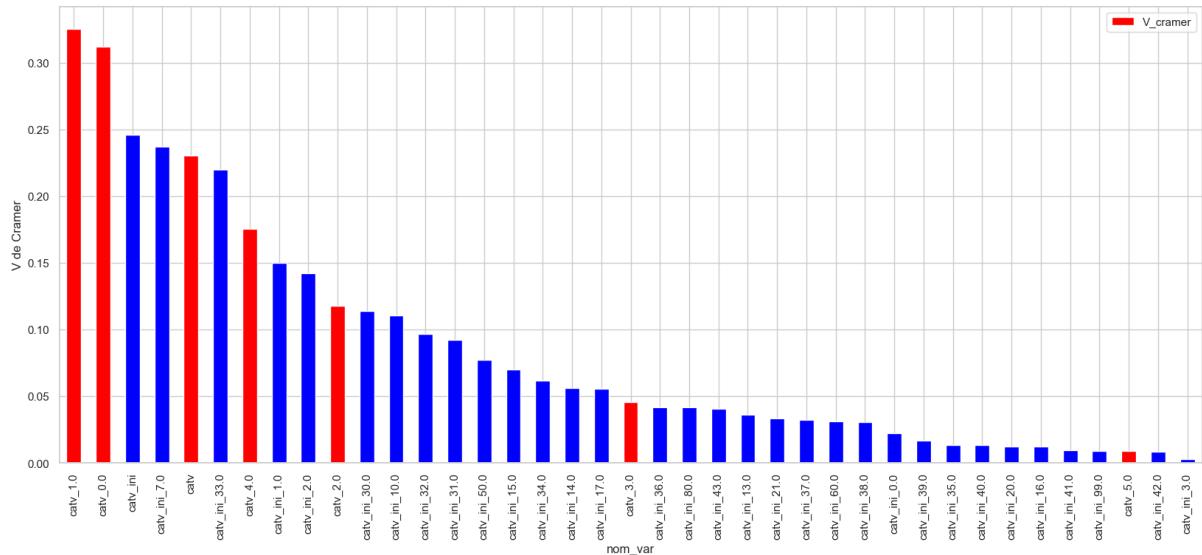


Figure 10 : Valeurs du V_Cramer pour les modalités de la catégorie de véhicule initiales et recodées

PLACE OCCUPÉE PAR L'USAGER et POINT DE CHOC INITIAL

Treatment des variables *catu*, *place*, *catv* et *choc*

Dans sa forme initiale, la base de données contient :

- Une variable **catu**, à 3 modalités, correspondant à la catégorie d'usager :
1 - *Conducteur*, 2 - *Passager*, 3 – *Piéton*
- Une variable **place**, à 10 modalités, permettant de situer la place occupée dans le véhicule par l'usager au moment de l'accident. La Figure 11 explique la catégorisation adoptée, la valeur 10, absente sur cette figure, étant associée aux piétons.

Transport en commun			
4	7	7	7
5	8	8	8
5	8	8	8
5	8	8	8
3	9	9	9

Figure 11 : Numérotation des places dans la variable *place*

- La catégorie de véhicules est connue grâce à la variable **catv**, à 5 modalités : 0 - Voitures, 1 - Motos, 2 - Poids lourds, 3 – Transports en commun, 4 – Vélos et trottinettes, 5 – Autres véhicules
- Une variable **choc**, à 10 modalités, informant sur le point de choc initial : 0 - *Aucun*, 1 - *Avant*, 2 - *Avant droit*, 3 - *Avant gauche*, 4 - *Arrière*, 5 - *Arrière droit*, 6 - *Arrière gauche*, 7 - *Côté droit*, 8 - *Côté gauche*, 9 - *Chocs multiples*

Ces variables ont été **recombinées** de façon à **limiter les modalités** (pour réduire la dimension du modèle), tout en conservant un maximum d'informations sur **la position de l'usager dans le véhicule** ainsi que sur **sa proximité avec le point de choc**. Ces deux aspects nous semblent en effet importants pour estimer la gravité d'un accident pour un usager donné. C'est pourquoi, nous avons réalisé les modifications suivantes :

- création d'une nouvelle variable proximité du point de choc,
- recodage de la variable place en 4 modalités.

Création d'une nouvelle variable binaire : prox_pt_choc

Une variable **prox_pt_choc**, binaire, est ainsi créée. Elle prend la valeur 1 lorsque l'usager est considéré à proximité du point de choc, 0 sinon (Tableau 2).

Tableau 2 : Configurations pour lesquelles la variable *prox_pt_choc* est prise égale à 1

Catégorie de véhicules	Point de choc initial	Place de l'usager
Voitures, Poids lourds, Transports en commun	1- Avant	1 – 6 – 2
	2- Avant droit	2
	3- Avant gauche	1
	4- Arrière	3 – 4 – 5
	5- Arrière droit	3
	6- Arrière gauche	6
	7- Côté droit	2 – 3 – 9
	8- Côté gauche	1 – 7 – 4
	9- Chocs multiples	Toutes les places
Motos, vélos et trottinettes	Toutes les places, quel que soit le point de choc	

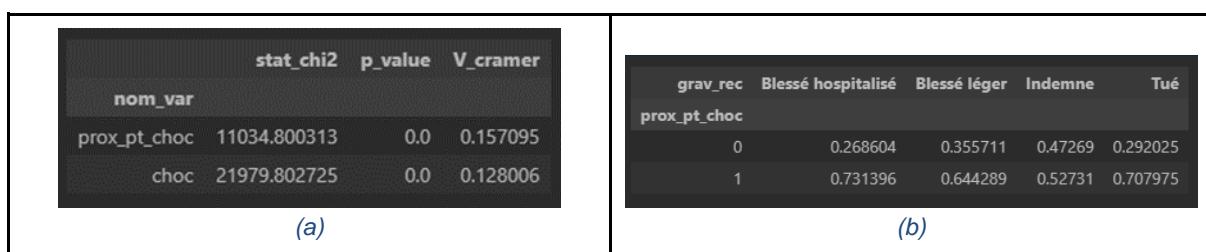


Figure 12 : (a). Comparaison du χ^2 et du V de Cramer pour les variables *prox_pt_choc* et *choc*, (b). Tableau de contingence de la variable *prox_pt_choc* avec la variable cible

Le test d'indépendance du χ^2 et le calcul du V de Cramer tendent à montrer une **relation plus importante de la variable cible avec la variable prox_pt_choc, qu'avec choc** (Figure 12 (a)). Le tableau de contingence (Figure 12 (b)) souligne que les proportions de blessés ou tués sont significativement plus importantes lorsque les usagers sont à proximité du choc.

Recodage de la variable place en 4 modalités

La Figure 13 montre que la variable “place” se répartit majoritairement sur les modalités conducteur, passager avant des véhicules ou arrière des motos, et piéton. Il a donc été choisi de rassembler certaines modalités pour en diminuer le nombre.

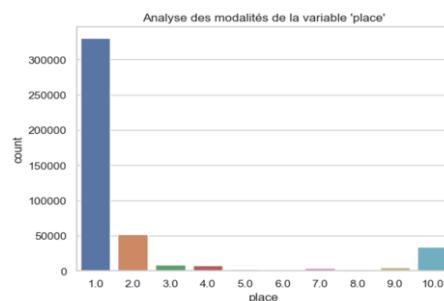


Figure 13 : Répartition des modalités de la variable *place*

La variable place est ainsi recodée pour apporter une précision supplémentaire à la variable catu, et la remplacer. La nouvelle variable place_rec contient 4 modalités : 1 – Conducteur, 2 – Passager avant, 3 – Passager arrière, 4 – Piéton. Le Tableau 3 donne les correspondances entre les places de la Figure 11 et les modalités de notre nouvelle variable.

Tableau 3 : Correspondances entre les modalités de la variable place

Modalités de la variable place	Voiture, Poids lourds, Transport en commun	1	2 - 6	3 à 9	10
	Moto	1		2 - 3	
Modalités de la nouvelle variable, place_rec	1 - Conducteur	1 - Conducteur	2 - Passager avant	3 - Passager arrière	4 - Piéton

Les tableaux des Figure 14 (a) et (b) soulignent que **le recodage permet de conserver les modalités les plus en relation avec la variable cible**, ce qui est prometteur quant à la réorganisation effectuée.

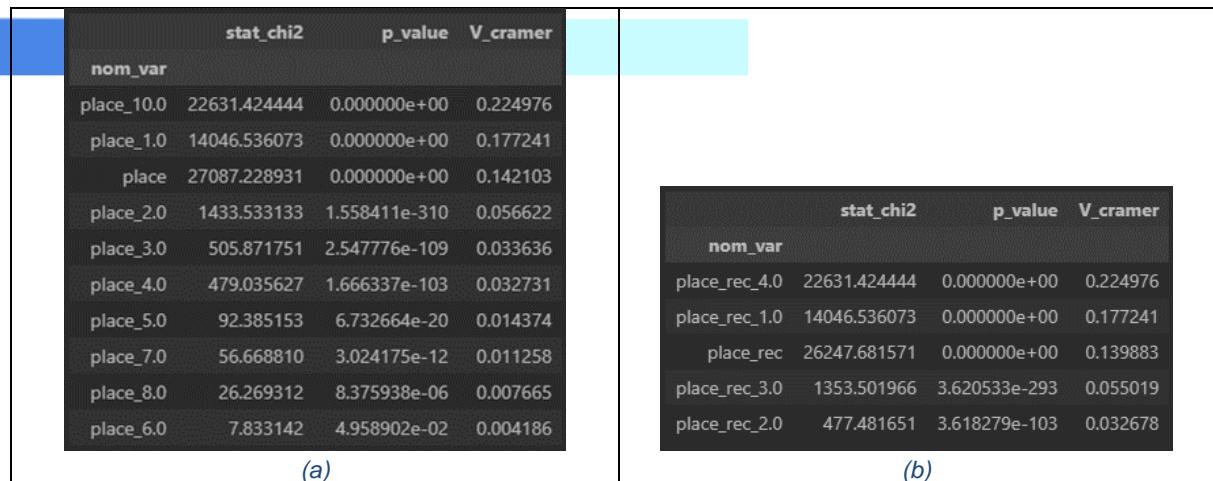


Figure 14 : χ^2 et V de Cramer pour les différentes modalités (a) de la variable place, (b). de la variable place_rec.

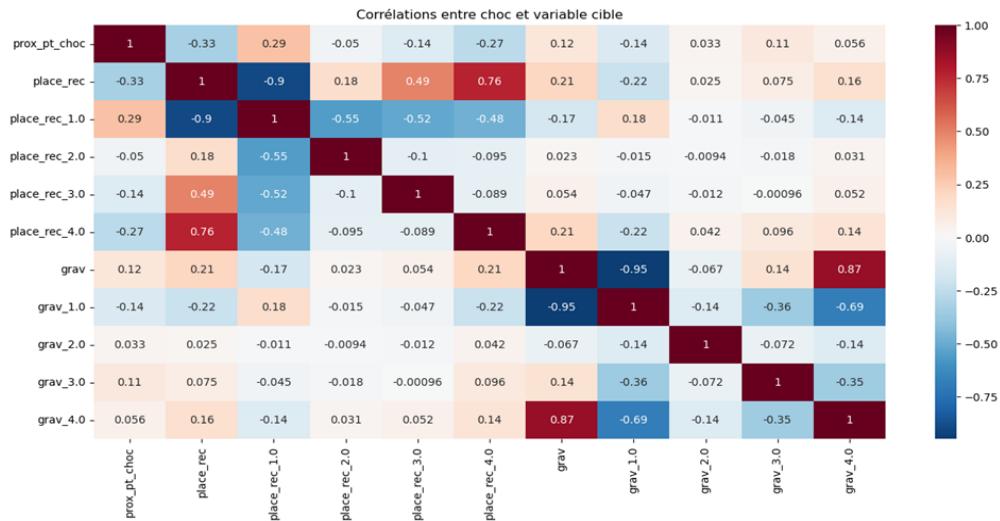


Figure 15 : Corrélation entre les différentes modalités de place_rec et de la variable cible. Les niveaux de gravité 1, 2, 3 et 4 correspondent respectivement à indemne, tué, blessé hospitalisé et blessé léger.

L'analyse des corrélations (Figure 15) souligne notamment que les piétons (à la place_rec 4) sortent rarement indemnes des accidents répertoriés dans les BAAC. Du côté des véhicules, la place de conducteur est négativement corrélée au fait d'être blessé léger et positivement corrélée au fait d'être indemne (protection de l'habitacle ?). Pour le reste des corrélations, les valeurs sont relativement faibles.

ÉLÉMENTS DE SÉCURITÉ

Traitement des variables secu1, secu2, secu3

La présence et l'utilisation d'éléments de sécurité par les usagers sont prises en compte dans la base de données par 3 variables distinctes : secu1, secu2, secu3. Secu1 renseigne sur le type d'un premier élément de sécurité, secu2 et secu3 informent sur un deuxième et troisième équipement le cas échéant. Chacune de ces variables a 10 modalités, recensées dans le tableau de la Figure 16.

En l'état, ces variables sont peu intéressantes car elles contiennent des informations identiques et ne peuvent être analysées indépendamment les unes des autres. Il a donc été choisi **de dichotomiser ces variables et de générer une nouvelle variable pour chaque équipement de sécurité, indiquant son utilisation, ou non, par l'usager impliqué dans l'accident.**

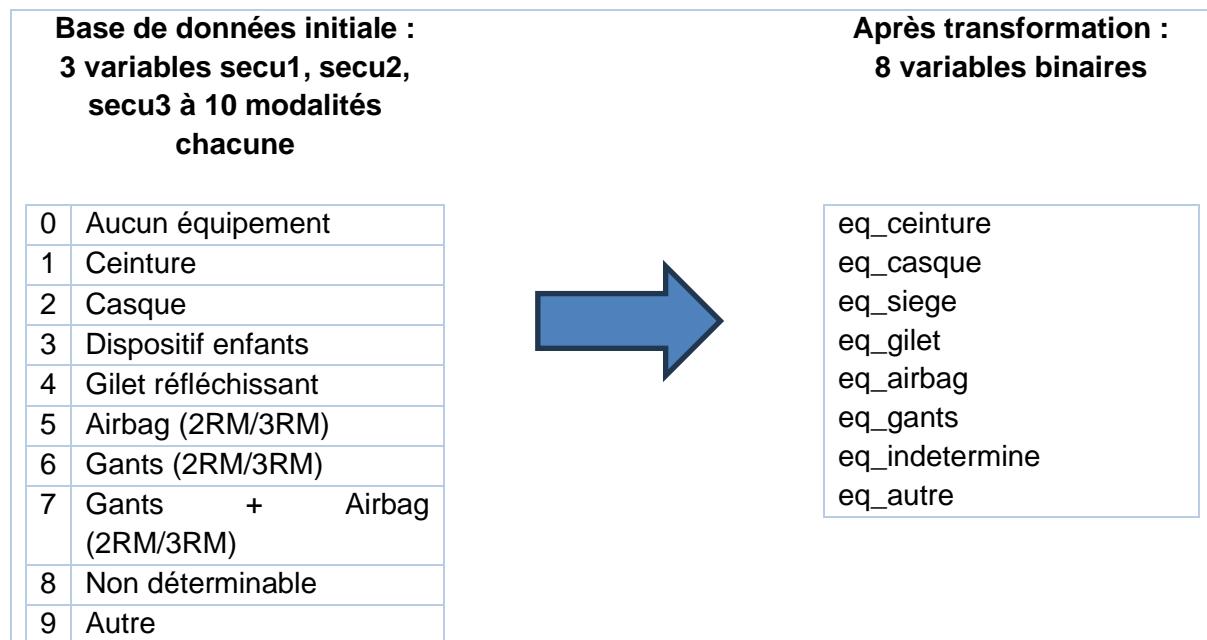


Figure 16 : Dichotomisation des équipements de sécurité

Cette variable recodée permet de mieux appréhender la gravité de l'accident en fonction du port ou non de l'équipement de sécurité. Le test d'indépendance du χ^2 sur les équipements de sécurité et sur la catégorie de véhicules (Figure 17) fait ressortir la significativité de ces variables et l'intensité relativement forte de leur relation avec la variable cible.

nom_var	stat_chi2	p_value	V_cramer
eq_ceinture	82249.401604	0.000000e+00	0.426129
eq_casque	49392.580652	0.000000e+00	0.330221
catv_1.0	48049.346525	0.000000e+00	0.325700
catv_0.0	44067.709326	0.000000e+00	0.311914
eq_gants	22104.529370	0.000000e+00	0.220910
catv_4.0	13916.418985	0.000000e+00	0.175282
catv_2.0	6286.570219	0.000000e+00	0.117810
eq_autre	5944.895508	0.000000e+00	0.114564
eq_indeetermine	1660.844001	0.000000e+00	0.060553
catv_3.0	949.501312	1.620146e-205	0.045785
eq_gilet	379.056727	7.608711e-82	0.028929
eq_airbag	240.205309	8.592272e-52	0.023029
eq_siege	186.454065	3.560829e-40	0.020289
catv_5.0	36.655608	5.441811e-08	0.008996

Figure 17 : χ^2 et V de Cramer pour les différentes variables équipement et catégorie de véhicule recodées

L'analyse des corrélations (Figure 18) montre que le port de la ceinture a une influence importante sur la variable cible, en étant positivement corrélée au fait d'être indemne, et négativement corrélée au fait d'être blessé (léger ou hospitalisé). Les résultats sont inversés sur le port du casque, ce qui est relativement contre-intuitif (plus de conduite à risque ?).

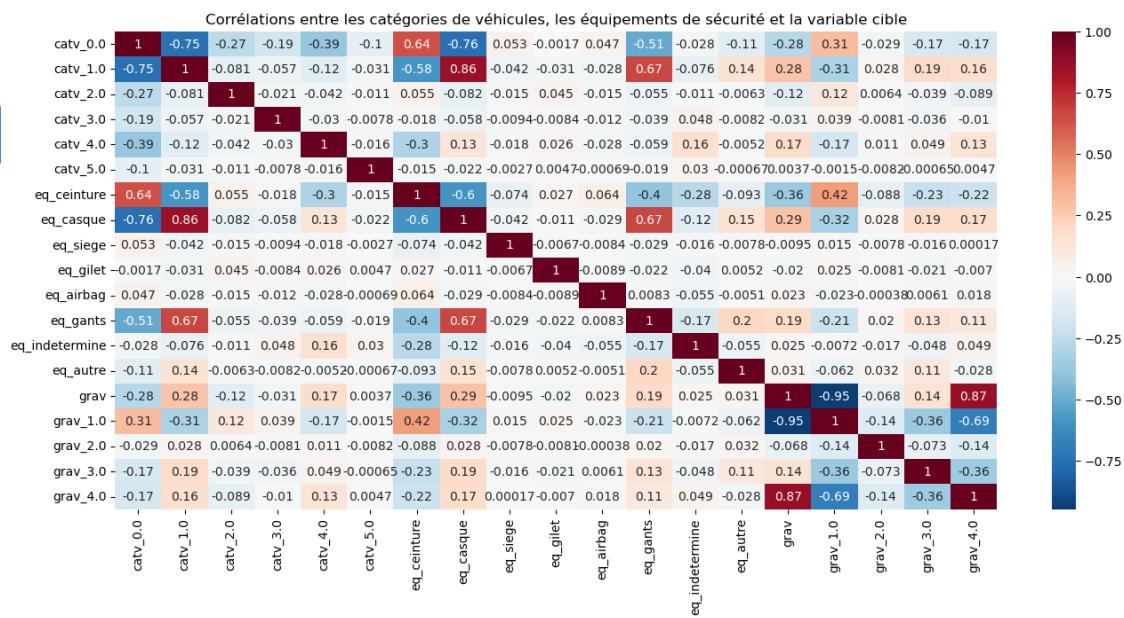


Figure 18 : Corrélation entre les différentes variables équipement, les différentes modalités de véhicule recodées et de gravité

Les variables sur les équipements sont assez logiquement fortement corrélées à certaines modalités des catégories de véhicules. Ainsi on observe :

- pour les gants et le casque : une corrélation positive avec la moto et une corrélation négative avec la voiture,
- pour la ceinture : une corrélation positive avec la voiture et une corrélation négative avec la moto

Cela permet aussi de connaître les équipements qui sont régulièrement utilisés ensemble :

- le casque a une corrélation positive avec les gants, et dans une moindre mesure, avec les équipements répertoriés ‘autre’ par les forces de l’ordre.

CONDITIONS ATMOSPHÉRIQUES

Traitement de la variable atm

La variable atm de la base de données initiale recense 9 modalités : 1 – Normale, 2 – Pluie légère, 3 – Pluie forte, 4 – Neige-grêle, 5 – Brouillard, fumée, 6 – Vent fort-tempête, 7 – Temps éblouissant, 8 – Temps couvert, 9 – Autre.

Cette variable traite à la fois de l'impact des conditions atmosphériques sur la tenue de route et sur la visibilité. La Figure 19 (a) montrant les proportions des états de gravité en fonction des modalités de cette variable, souligne l'impact des conditions atmosphériques sur la gravité, confirmé par des p-valeurs au test d'indépendance du χ^2 inférieures à 5%. En revanche, les intensités des relations de ces variables avec la variable cible, mesurées avec le V de Cramer (Figure 19 (b)), s'avèrent très faibles. Les modalités “temps éblouissant” et “pluie légère” apparaissent comme les plus influentes. Mais ces modalités nous semblent largement soumises à l'interprétation des forces de l'ordre : un temps éblouissant pour un officier pouvant apparaître comme un temps normal pour un autre. Il en est de même pour les modalités pluies, temps couvert, etc.

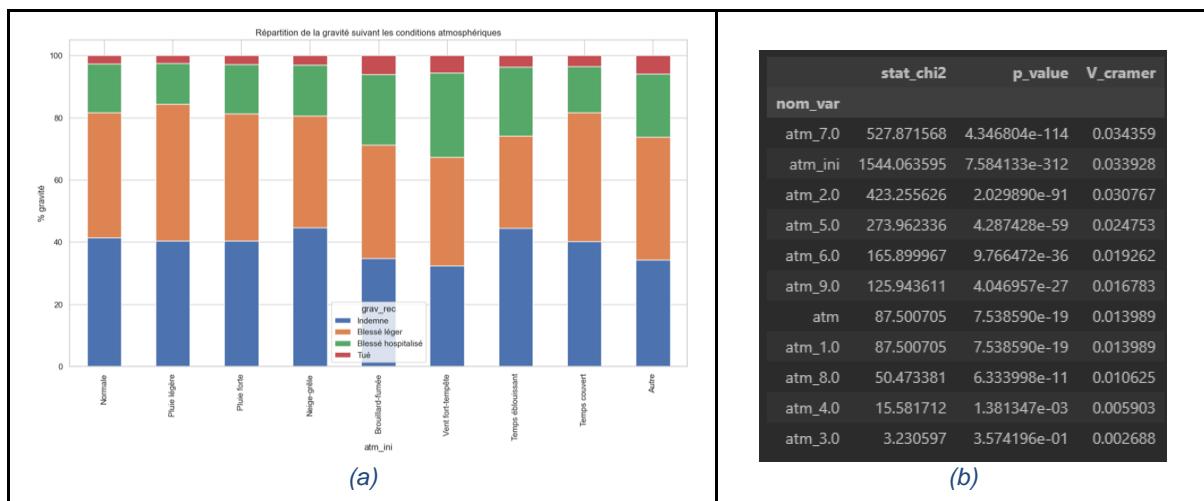
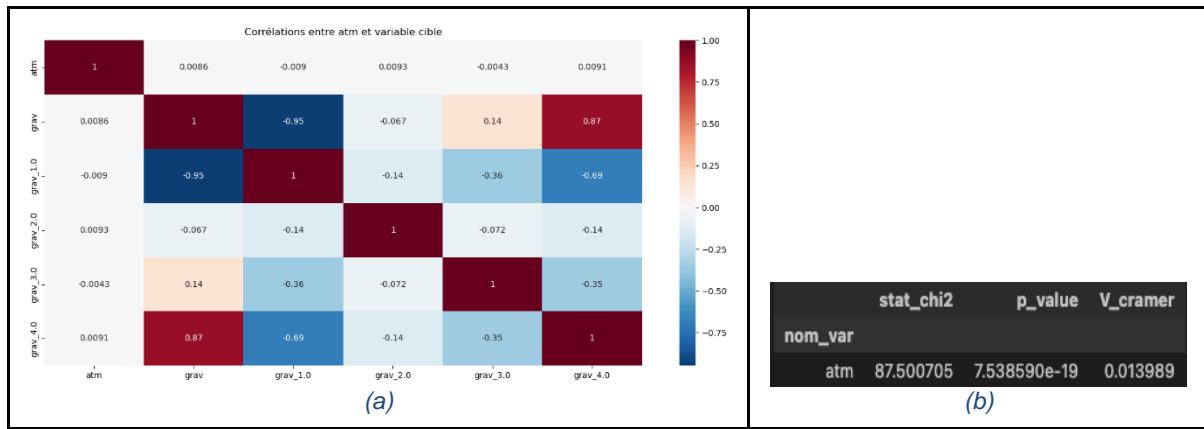


Figure 19 : (a). Proportion de la gravité selon les conditions atmosphériques, (b). χ^2 et V de Cramer pour les différentes modalités de la variable atm

Ce biais potentiel dans le remplissage des bulletins nous a conduits à ne pas souhaiter multiplier les modalités sur cette variable et nous avons opté pour un ré-encodage binaire de la variable atm, prenant désormais 0 en *conditions normales*, et 1 en *conditions dégradées* (pluie légère, pluie forte, neige-grêle, brouillard-fumée, vent fort-tempête, temps éblouissant, temps couvert et autre).

La corrélation entre la variable ‘atm’ recodée en binaire et la variable cible indique une influence des conditions atmosphériques sur la survenue d'accidents avec des blessés hospitalisés ou tués (Figure 20 (a)). Cependant, le V de Cramer indique une faible influence de cette variable sur notre variable cible (Figure 20 (b)).



MANŒUVRE PRINCIPALE AVANT L'ACCIDENT

Traitement de la variable manv

La variable ‘manv’ de la base de données initiale recense 27 modalités ce qui ne permet pas de l’utiliser ainsi.

Nous avons donc effectué des regroupements, tels que présentés sur la Figure 21. La variable *manv* réencodée présente 4 modalités : 0 – *Même sens*, 1 – *Contresens*, 2 – *Immobile*, 3 – *Changement de direction*.



Figure 21 : Regroupement des catégories pour la variable *manv*

Le recodage de la variable en 4 modalités permet d'avoir une meilleure vision de l'impact du mouvement du véhicule au moment de l'accident. Ainsi, on peut constater que les accidents les plus graves surviennent lorsque le véhicule roule à contresens (Figure 22). Réciproquement, les accidents les moins graves interviennent lorsque le véhicule est à l'arrêt.

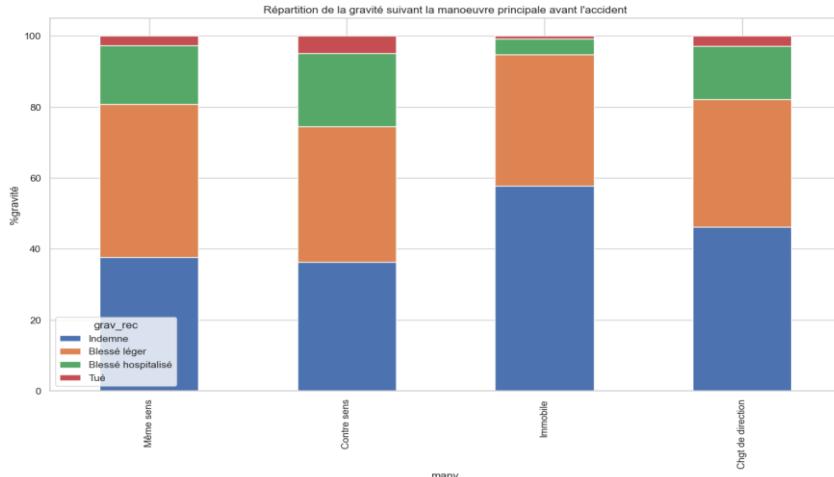


Figure 22 : Proportion de la gravité selon la manœuvre principale avant l'accident

L'étude de la corrélation entre la manœuvre au moment de l'accident et la gravité de l'accident (Figure 23 (a)) confirme les observations précédentes, mais montre aussi une faible corrélation en général de cette variable avec la variable cible.

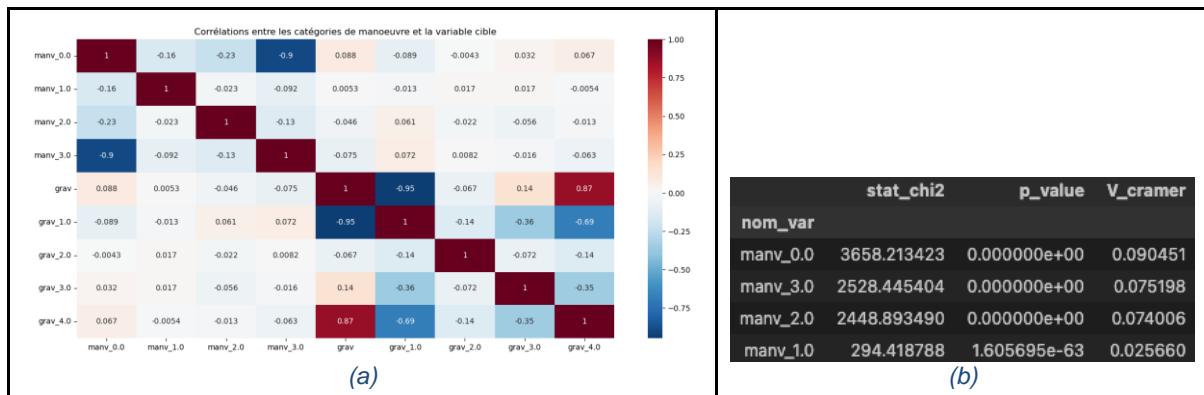


Figure 23 : (a). Corrélation entre les différentes modalités de la variable manv et de la variable cible, (b). χ^2 et V de Cramer pour les différentes modalités de la variable manv

Le test d'indépendance du χ^2 (Figure 23 (b)) montre des p_valeurs inférieures à 5% confirmant l'impact de la manœuvre sur la gravité de l'accident. En revanche, les valeurs du V de Cramer, indiquant les intensités des relations de ces variables avec la variable cible, sont relativement faibles. Nous décidons de garder cette variable dans une première approche du modèle quitte à la supprimer par la suite.

I.5 Analyse des variables continues

Les variables continues de la base de données sont au nombre de 5 : age_usager, latitude, longitude, mois et heure.

En fonction des modélisations envisagées, notamment si elles ont recours ou non à des calculs de distance, il peut être nécessaire de normaliser/standardiser les variables continues. Pour chacune des variables continues de notre base de données, il est précisé ci-après quel procédé pourra être utilisé en cas de besoin d'une normalisation, et les raisons de ce choix.

1.5.1 Age_usager

La variable age_usager ne suit pas une distribution normale (Figure 24 (a) et (b)) et présente quelques outliers (Figure 24 (c)) qui restent cependant dans des ordres de grandeur admissibles comparativement aux autres valeurs. **En cas de besoin, une normalisation min-max pourra être envisagée pour cette variable.**

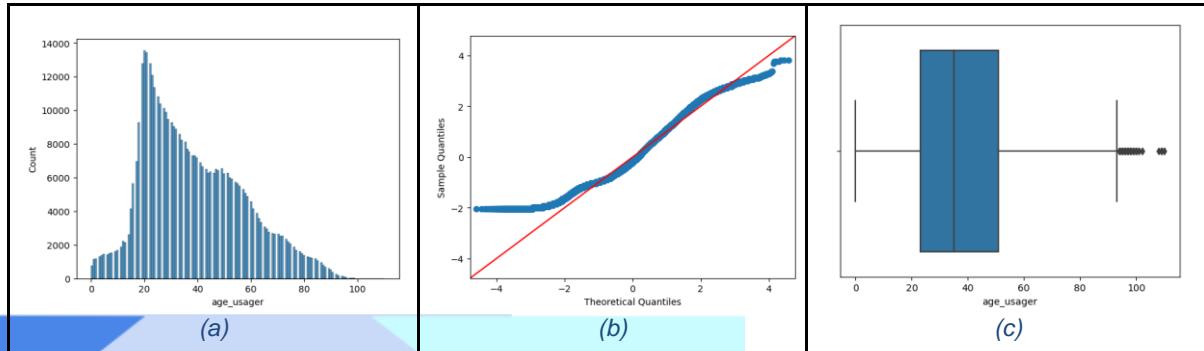


Figure 24 : Distribution de age_usager. (a). Histogramme des valeurs, (b). Graphique des quantiles, (c). Boîte à moustaches

1.5.2 Heure et mois

Il est recommandé dans les forums dédiés à la data science (Kaleko, 2017) de procéder à des transformations sinus/cosinus pour les variables temporelles de type heure et mois. En effet, ces modifications permettent de conserver le côté cyclique de ces variables temporelles.

$$\begin{cases} \text{heure}_{\sin} = \sin\left(2\pi \frac{\text{heure}}{24}\right) & \text{mois}_{\sin} = \sin\left(2\pi \frac{\text{mois}-1}{12}\right) \\ \text{heure}_{\cos} = \cos\left(2\pi \frac{\text{heure}}{24}\right) & \text{mois}_{\cos} = \cos\left(2\pi \frac{\text{mois}-1}{12}\right) \end{cases}$$

1.5.3 Latitude et longitude

Les variables latitude et longitude ne suivent pas des distributions normales (Figure 25) et présentent de nombreux outliers (Figure 26) en raison de la présence des DOM/TOM dans la base de données. **En cas de besoin d'une normalisation, le recours à un procédé de Robust Scaling est envisagé pour ces variables.**

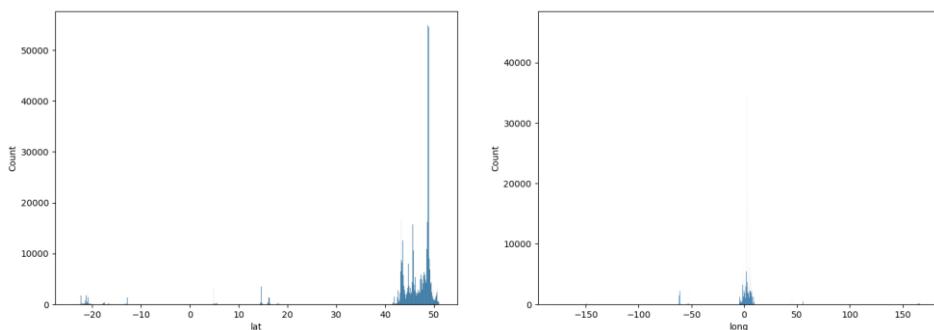


Figure 25 : Distributions des variables lat et long

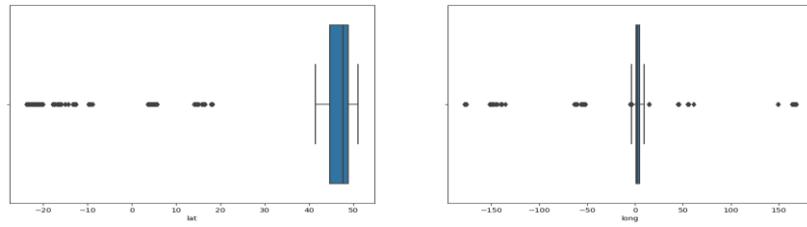


Figure 26 : Boîtes à moustaches des variables lat et long

En se focalisant uniquement sur la métropole, on observe toujours une distribution qui n'est pas normale (Figure 27) et la présence d'outliers (Figure 28). Il faudrait aussi avoir recours à un procédé de Robust Scaling dans le cas où une normalisation serait nécessaire.

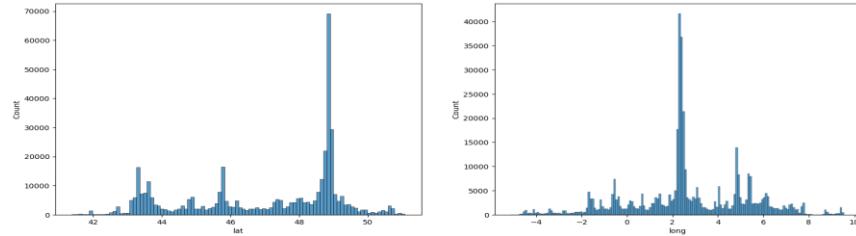


Figure 27 : Distributions des variables lat et long, pour la métropole uniquement

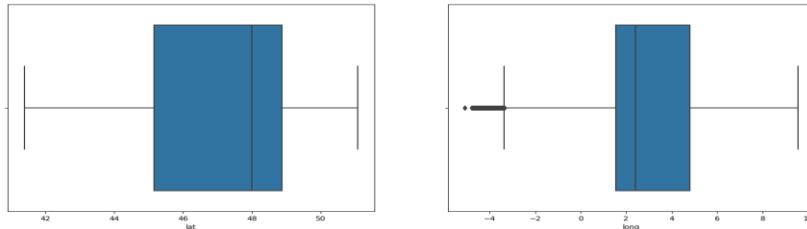


Figure 28 : Boîtes à moustaches des variables lat et long pour la métropole uniquement

I.6 Visualisations et Statistiques

I.6.1 Evolution Temporelle

A. Mois

On voit se dessiner une certaine saisonnalité dans la proportion mensuelle de tués et de blessés hospitalisés (Figure 29), avec plus de gravité chaque été, et un pic lors du premier confinement (avril 2020).

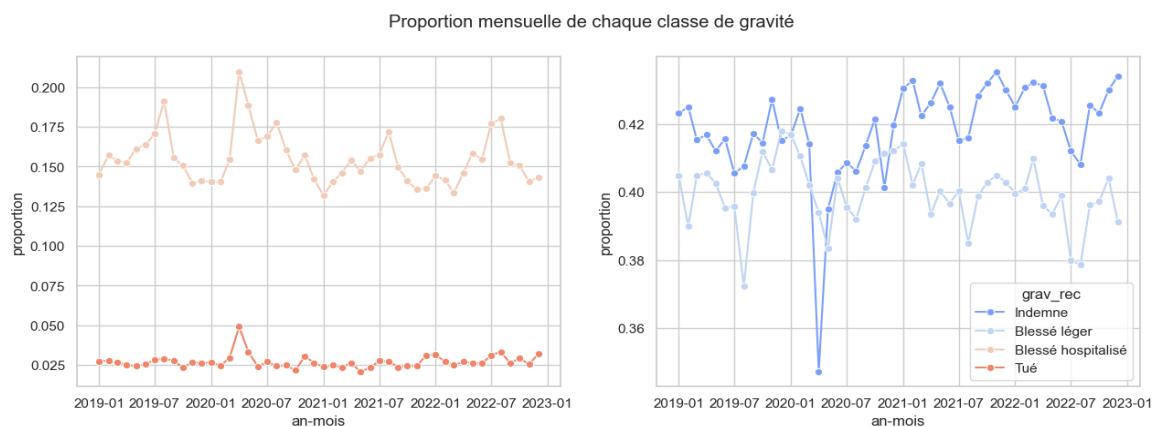


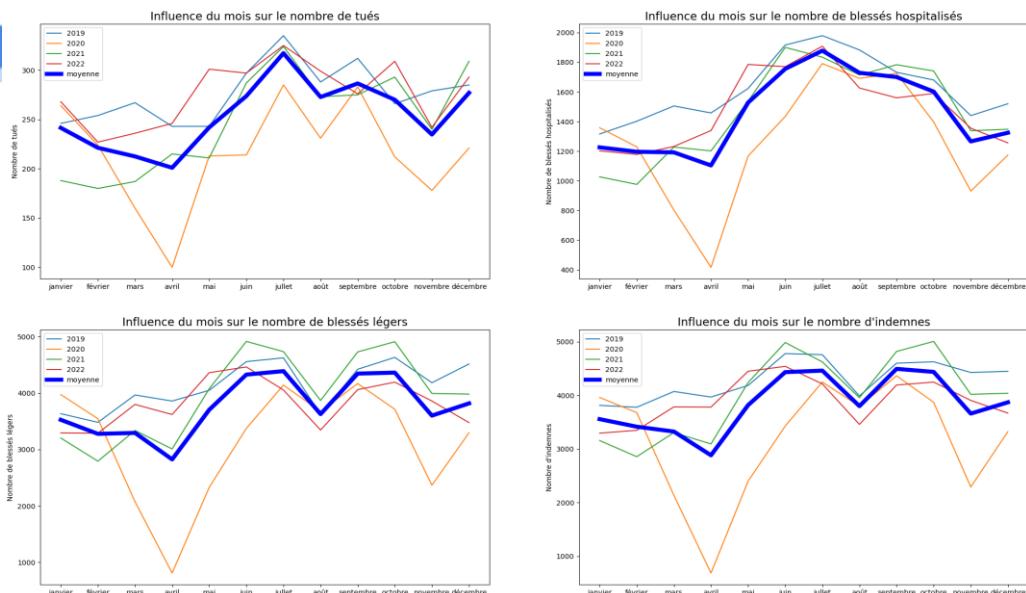
Figure 29 : Proportion mensuelle de chaque classe de gravité

Toutefois, dans un premier temps, nous allons axer la modélisation sur des méthodes de classification et par conséquent laisser de côté la variable “an-mois” au profit d’une variable “mois de l’année”.

En comparant la gravité des accidents mensuels sur les 4 années (Figure 30), on peut se rendre compte d’une saisonnalité :

- diminution du nombre de tués, blessés légers et usagers indemnes au mois d’août par rapport aux mois de juillet et septembre, peut-être due à la période estivale où la proportion de personnes en vacances est la plus importante,
- diminution de toutes les modalités de gravité aux mois d’avril et de novembre (correspondant aux vacances scolaires ?)
- augmentation de toutes les modalités de gravité aux mois de juin/juillet.
- augmentation du nombre de tués, blessés légers, usagers indemnes (et de blessés hospitalisés certaines années) aux mois de septembre/octobre.

L’année 2020 présente 2 pics bien en dessous des autres années correspondant aux 2 périodes de confinement. Le premier pic est plus important car il correspond au confinement strict pour la période du 17 mars au 11 mai 2020. Le second est moins accentué car la période de confinement du 30 octobre au 15 décembre 2020 était moins stricte.



La gravité subit donc de fortes variations selon les mois. Nous décidons donc de la conserver sans créer de catégories. En revanche, nous observons des diminutions sur les courbes pour les mois d’avril, d’août et de novembre qui pourraient correspondre à des périodes de vacances scolaires. Nous décidons donc de créer une variable jour_chome qui regroupe les jours fériés et les vacances scolaires (Figure 31).

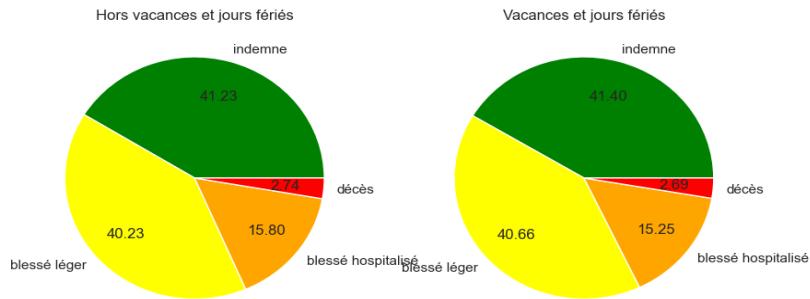


Figure 31 : Proportion des différentes modalités de gravité selon s'il s'agit d'un jour de vacances et jours fériés ou non

L'étude des statistiques du χ^2 et du V de Cramer (Figure 32) montre que la gravité est bien dépendante de la variable jour_chome mais qu'elle influe très faiblement sur la gravité. Nous conservons cette variable dans un premier temps.

	stat_chi2	p_value	V_cramer
nom_var			
jour_chome	21.745782	0.000074	0.006974

Figure 32 : χ^2 et V de Cramer pour la variable jour_chome

B. Jour

Le jeu de données initial permet de connaître la date précise de l'accident grâce aux variables 'an', 'mois' et 'jour' et de créer la variable 'jour_semaine'. Nous avons pu ainsi retrouver le jour de la semaine où l'accident avait eu lieu. L'étude de la gravité en fonction du jour de la semaine (Figure 33) permet de visualiser une différence entre les jours de la semaine :

- un nombre relativement constant des modalités de gravité du lundi au jeudi,
- une augmentation de toutes les modalités de gravité le vendredi (due aux départs en week-end?),
- une augmentation du nombre de tués et blessés hospitalisés et, de manière concomitante, une diminution du nombre de blessés légers et usagers indemnes les samedis et dimanches (due à la pratique de loisirs?).

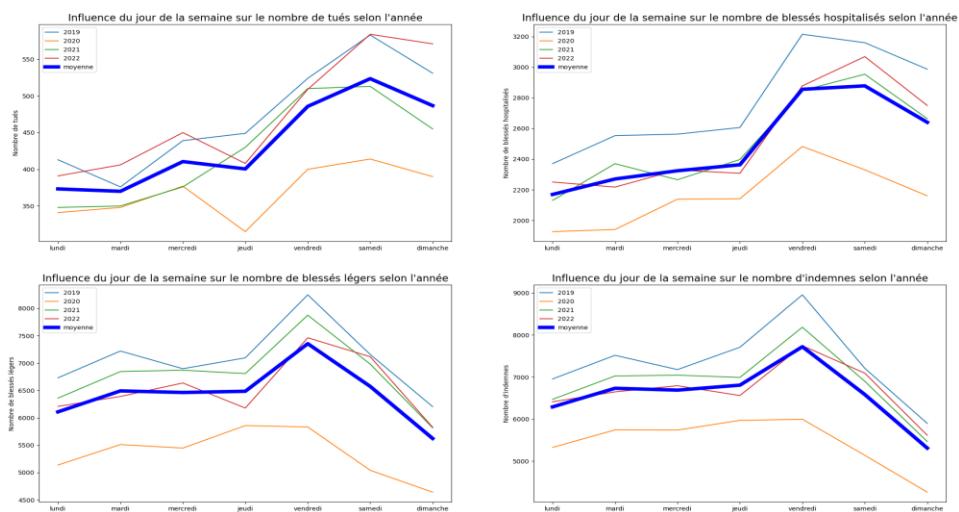


Figure 33 : Courbes du nombre d'usagers pour chaque classe de gravité selon le jour de la semaine pour les années 2019 à 2022

Suite à ces observations, un regroupement en 2 modalités nous semble intéressant : jours de semaine et week-end. En revanche, nous nous demandons s'il est intéressant d'inclure ou non le vendredi dans le weekend. Pour répondre à cela, nous avons comparé les proportions de chaque gravité dans le cas d'une variable binaire weekend (Figure 34 (a)) et d'une variable binaire vendredi et weekend (Figure 34 (b)). La proportion de tués passe de 2.43% à 3.20% le week-end, et celle de blessés hospitalisés de 14.49 à 17.78%, alors que en incluant vendredi dans le week-end, la proportion de tués passe de 2.40% à 2.95%, et celle de blessés hospitalisés de 14.28 à 16.81%. Comme la différence entre les modalités 0 et 1 étant plus importantes pour les blessés hospitalisés et les tués dans le cas de la variable weekend sans le vendredi, nous estimons que cette variable est le meilleur choix.

grav_rec	Indemne	Blessé léger	Blessé hospitalisé	Tué
weekend				
0	42.894608	40.182603	14.489761	2.433029
1	39.555243	39.465550	17.779919	3.199288

(a)

grav_rec	Indemne	Blessé léger	Blessé hospitalisé	Tué
vendredi_weekend				
0	42.953148	40.369549	14.278298	2.399005
1	40.737317	39.495822	16.812417	2.954445

(b)

Figure 34 : Tableaux de contingence pour la variable weekend (a) excluant le vendredi, (b) incluant le vendredi

Pour confirmer ces résultats, nous réalisons un test du χ^2 et calculons la valeur du V de Cramer pour ces deux variables (Figure 35 (a) et (b)). On remarque que dans le cas de la variable vendredi_weekend la p_valeur du test du χ^2 est supérieure à 5%, donc on ne peut pas rejeter que la variable vendredi_weekend soit indépendante de la gravité. Nous gardons donc la variable weekend qui a une p_valeur inférieure à 5%.

stat_chi2	p_value	V_cramer
nom_var		
weekend	1057.571219	5.832204e-229

(a)

stat_chi2	p_value	V_cramer
nom_var		
vendredi_weekend	2.605329	0.456556

(b)

stat_chi2	p_value	V_cramer
nom_var		
jour_1	43.835374	1.635726e-09
jour_3	14.384992	2.425307e-03
jour_0	6.041107	1.096273e-01
jour_6	5.787469	1.224208e-01
jour_4	4.658266	1.986004e-01
jour_5	2.636677	4.510958e-01
jour_2	1.863872	6.011352e-01

(c)

Figure 35 : χ^2 et V de Cramer pour les variables (a). week-end sans le vendredi, (b). weekend avec le vendredi, (c) jour_semaine

De plus, en comparant avec la variable jour_semaine initiale, on observe que la valeur du V de Cramer est nettement améliorée dans le cas de la variable weekend (V_Cramer de 0,048) par rapport à la variable jour_semaine dummisée pour chaque jour de la semaine (Figure 35 (b) et (c)). Le choix de regrouper les jours en weekend ou non nous permet donc d'obtenir une variable avec une relation à plus forte intensité avec la gravité qu'en prenant chaque jour de la semaine séparément.

Au final, nous obtenons une répartition de la gravité selon les jours du lundi au vendredi et le week-end (Figure 36).

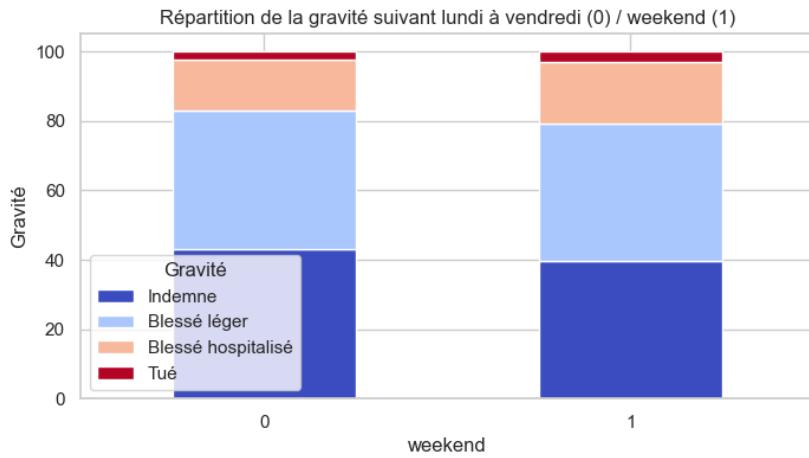


Figure 36 : Répartition de la gravité pour la variable weekend

C. Heure

Enfin, en récupérant l'heure de l'accident, en l'extrayant de la variable 'hrmn', nous étudions l'influence de l'heure de l'accident sur la gravité (Figure 37). Dans tous les cas de gravité, on observe un pic autour de 17h, s'étalant de 13h à 22h environ, ce qui correspond certainement à une augmentation du trafic routier l'après-midi. De même, le nombre diminue la nuit entre 22h et 4h correspondant aux heures où la circulation est la plus faible. En revanche, on note un léger pic des blessés légers et des usagers indemnes vers 8h, tandis que le nombre de tués augmente progressivement de 4h à 11h.

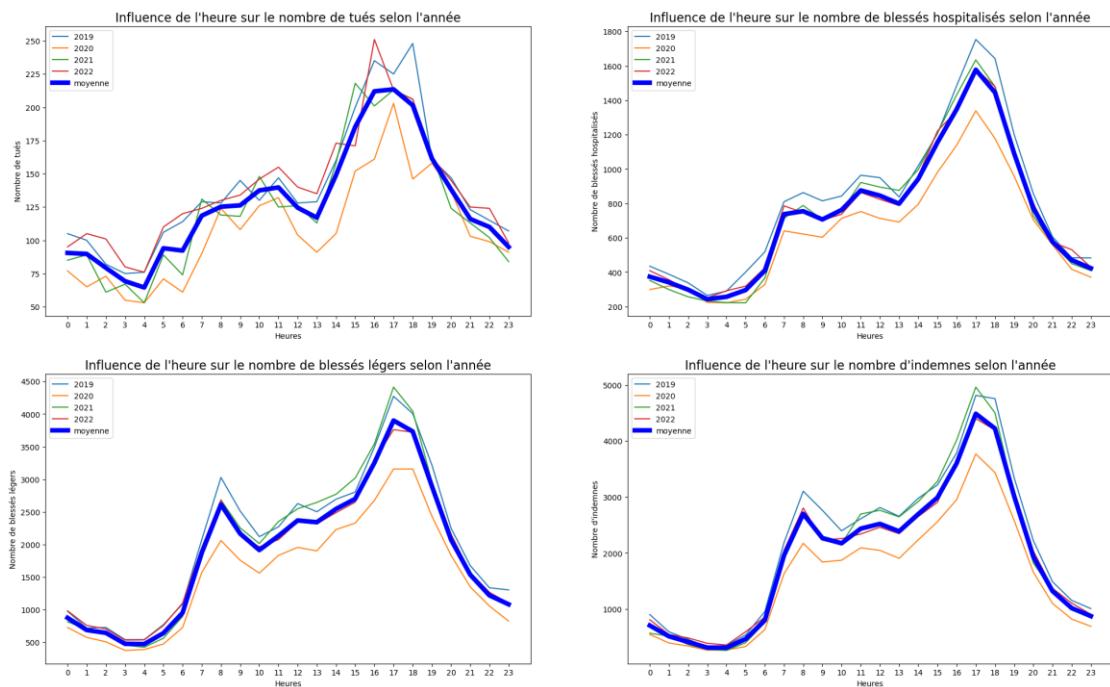


Figure 37 : Courbes du nombre d'usagers pour chaque classe de gravité selon l'heure pour les années 2019 à 2022

Ces variations différentes selon les gravités et l'évolution continue selon les heures pour une même gravité nous amène à conserver l'heure sans la catégoriser.

I.6.2 Disparités spatiales

A. Localisation des accidents en France métropolitaine

La précision du jeu de données avec les latitudes et longitudes permet de placer précisément la localisation des accidents avec leur gravité (Figure 38). Cette visualisation montre que les accidents ont plus fréquemment lieu au niveau des grandes agglomérations (Paris, Lyon, Lille, Marseille, Bordeaux...) mais aussi sur les principaux axes routiers (on reconnaît facilement le tracé de l'autoroute du Soleil par exemple). Enfin, les routes de la côte méditerranéenne semblent plus propices aux accidents. Ceci peut s'expliquer par la densité de population dans les agglomérations et la fréquentation plus importante des grands axes (notamment la zone méditerranéenne avec l'héliotropisme lors des vacances).

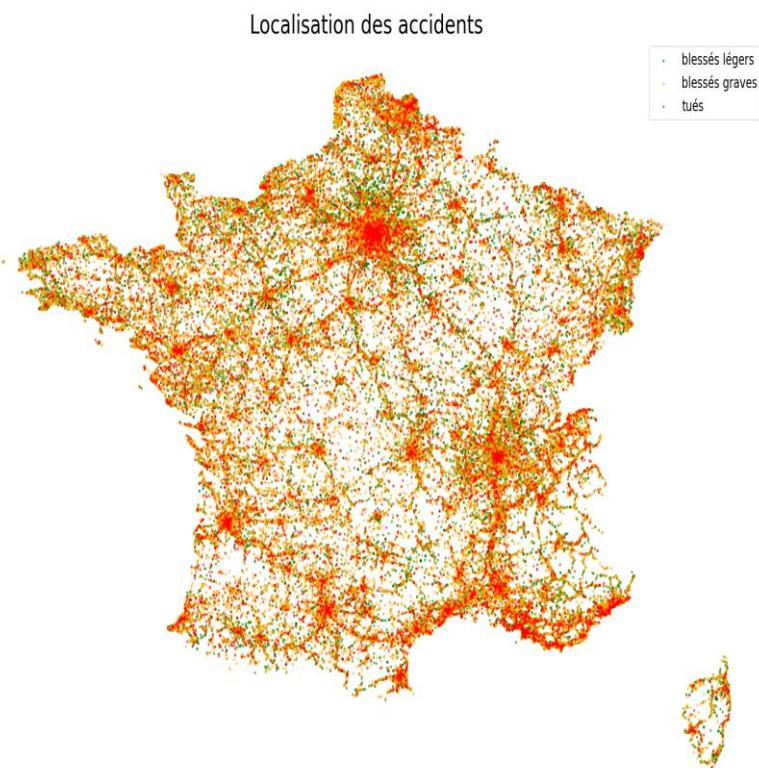


Figure 38 : Carte de la localisation des accidents selon la gravité en France métropolitaine

Si cette carte est révélatrice de la localisation des accidents, elle ne reflète cependant pas la proportion de la gravité des accidents en fonction du nombre d'accidents. C'est pourquoi nous nous sommes intéressés à la gravité des accidents selon la localisation.

B. Proportion de gravité des accidents selon la localisation

La Figure 39 (a) s'intéresse aux variations régionales dans la répartition des accidents. La taille des camemberts est directement indexée sur le nombre d'accidents par région, tandis que les portions de camembert renseignent sur les différents états de gravité des accidents. Ainsi, il apparaît que si la majorité des accidents se produisent en Ile de France, ce n'est pas dans cette région qu'ils sont les plus meurtriers. Les départements et territoires d'outre-mer, la Bourgogne-Franche-Comté, par exemple, se révèlent en proportion plus touchés par la mortalité routière. Si l'on regarde à l'échelle des départements (Figure 39 (b)), certains

départements (Les Landes, la Haute Saône) enregistrent des proportions de décès plus importantes que les autres. Il serait intéressant de voir si les modèles permettront d'identifier quelles spécificités présentent ces départements pour comprendre ces disparités géographiques.

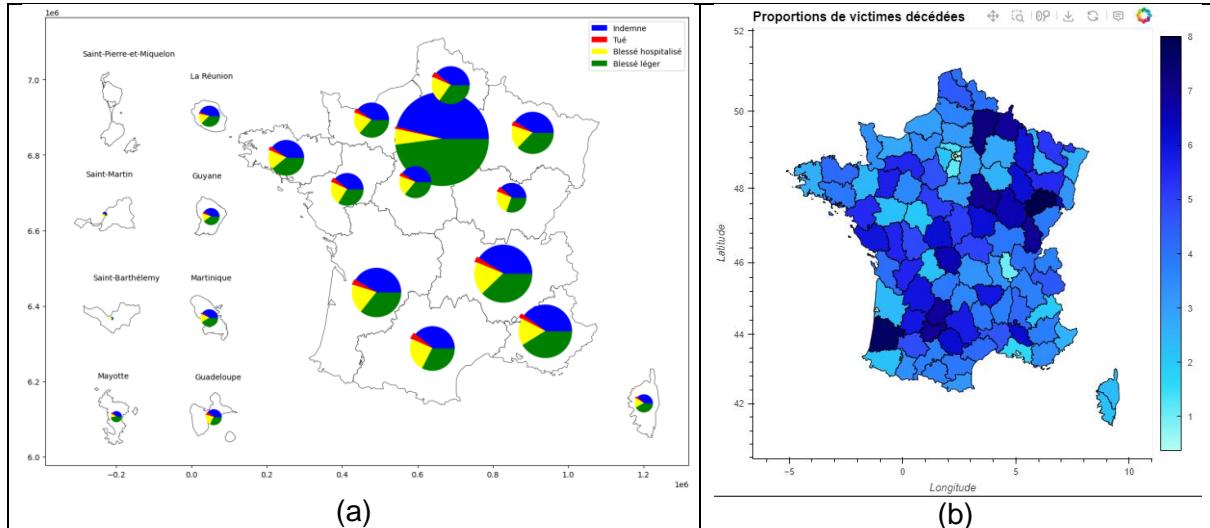


Figure 39 : Cartes (a) de la répartition des états de gravité des accidents par région (La taille des camemberts est proportionnelle au nombre d'usagers impliqués dans les accidents de chaque région), (b) des proportions de victimes décédées selon le département de France métropolitaine.

I.6.3 Influence de l'âge et du sexe dans l'accidentalité routière

A. Influence de l'âge sur la gravité

La Figure 40 présente la répartition des états de gravité en fonction de l'âge des usagers impliqués. Il est intéressant de noter sur cette figure que :

- pour les moins de 18 ans, la part de blessés (légers et hospitalisés) augmente dans les accidents avec l'âge,
- la gravité de l'accident tend à être plus sévère lorsque l'usager est âgé. La part de tués (en rouge sur la figure) augmente de manière importante à partir de 60 ans environ, aux dépens de la part des usagers indemnes.

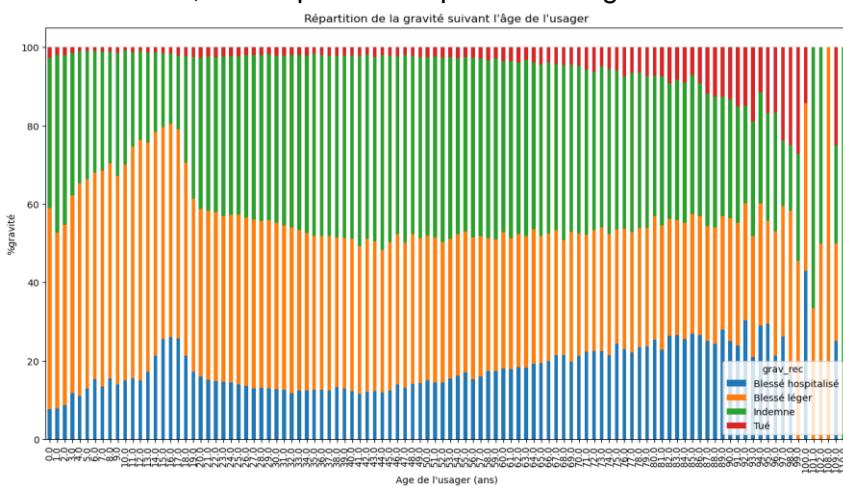


Figure 40 : Graphique de la répartition des modalités de la gravité en fonction de l'âge

Pour la modélisation, deux possibilités ont été envisagées : scinder cette variable en classes d'âges avec des bornes que nous aurions fixées, ou conserver cette variable en tant que variable continue. La seconde option a été choisie pour la modélisation de façon à ne pas perdre d'informations en choisissant nous-mêmes des bornes, qui n'auraient donc pas été

optimales. Des analyses par arbres de décision nous permettront peut-être de faire ressortir des bornes d'âge plus pertinentes que celles que nous aurions choisies initialement.

Pour l'analyse des dépendances entre variables en revanche, des classes d'âge ont été définies, par tranche de 5 ans.

B. Influence du sexe sur la mortalité lors d'un accident routier

La Figure 41 analyse la mortalité routière en fonction de l'âge et du sexe des usagers. La **classe d'âge des 20-24 ans est la plus impactée**, quelle que soit le sexe de l'usager. On recense au total 1433 tués dans cette tranche d'âge, contre 609 en moyenne, toutes tranches d'âges confondues. Du côté des hommes, les classes de 15 à 34 ans ont des taux de mortalité élevés. Ce taux tend à diminuer à mesure que l'âge augmente. En revanche, du côté des femmes, les 15 à 25 ans, mais également les plus âgées (au-delà de 70 ans) sont celles où l'on observe des proportions de décès plus importantes du côté des femmes que du côté des hommes.

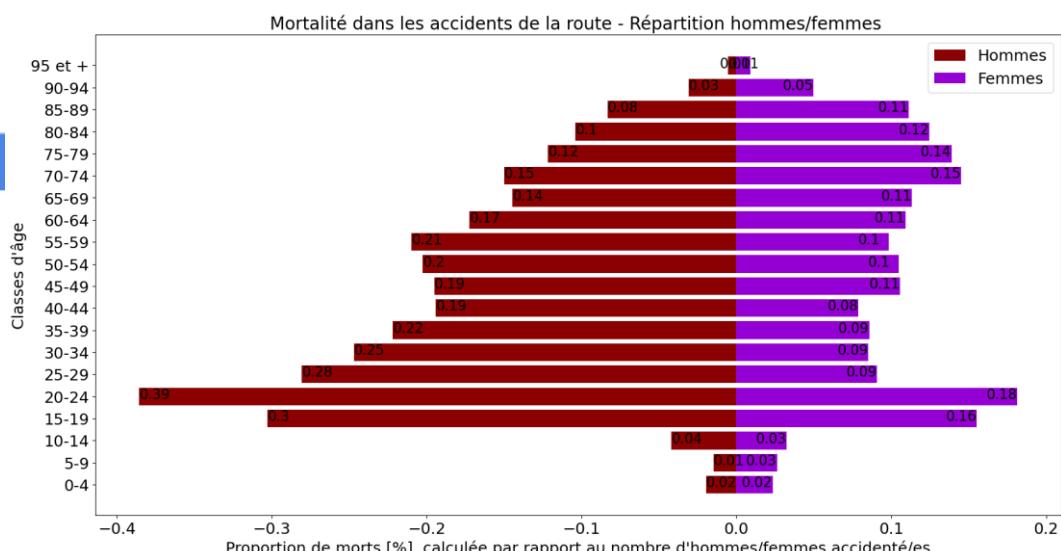


Figure 41 : Graphique de la mortalité des accidents de la route selon le sexe. Le calcul des proportions se base sur 9556 hommes tués, contre 2645 femmes (soit une mortalité masculine 3 à 4 fois supérieure).

C. Influence de la catégorie de véhicule selon l'âge sur la mortalité

La Figure 42 souligne que la voiture est le principal mode de déplacement impliqué dans les accidents, quelle que soit la catégorie d'âge. Viennent ensuite les motos pour les usagers de moins de 70 ans, et les vélos pour les plus de 70 ans. On note que les transports en commun et les poids lourds représentent une très faible partie des proportions de tués.

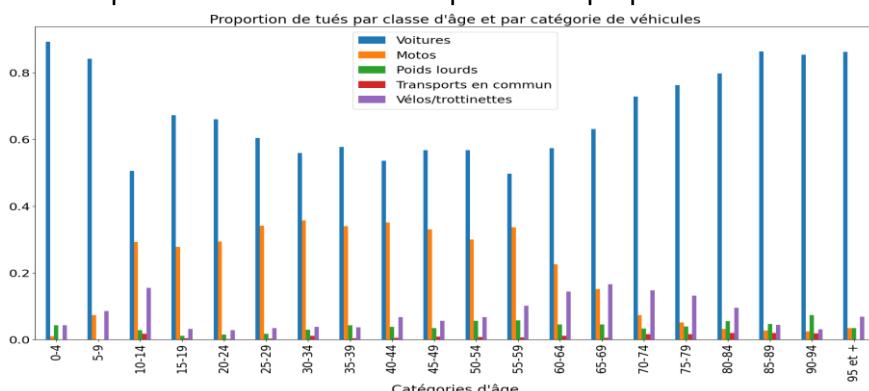


Figure 42 : Proportions, par classes d'âge, des catégories de véhicules associées aux usagers décédés

D. Comparaison des places occupées selon le sexe dans les accidents

Sur 100 hommes impliqués dans des accidents, 80 sont conducteurs, 7 sont passagers avant, 7 sont passagers arrière et 6 sont piétons. Sur 100 femmes impliquées dans des accidents, 59 sont conductrices, 16 sont passagères avant, 13 sont passagères arrière et 12 sont piétonnes. On observe donc une disparité importante entre les hommes et les femmes dans les places occupées dans les véhicules (Figure 43).

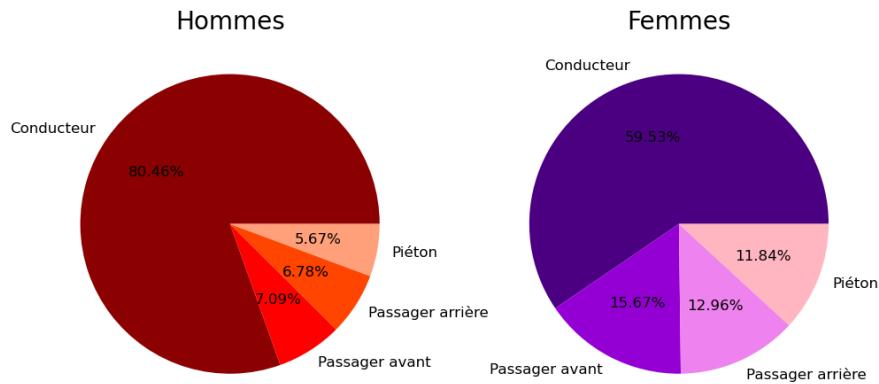


Figure 43 : Comparaison de la position de la personne accidentée selon le sexe

I.7 Conclusion

Le problème de prédiction de la gravité des accidents routiers est donc un problème de classification supervisée sur un jeu de données déséquilibré, issu de la base de données des accidents de la route administrée par l'ONISR.

Les grandes étapes de pre-processing et feature engineering ont consisté à :

- Restreindre notre champ d'étude aux années 2019 à 2022,
- Supprimer les doublons,
- Analyser les valeurs manquantes pour supprimer des observations, ou des variables selon l'ampleur du manque d'informations,
- Analyser les relations entre chaque variable et la variable cible pour, selon les cas :
 - Supprimer la variable (ex : numéro du PR de rattachement),
 - Regrouper certaines modalités de la variable, tout en conservant un maximum de relation avec la variable cible (ex : catégorie du véhicule),
 - Conserver la variable telle quelle (ex : type de collision)
- Créer de nouvelles variables, potentiellement intéressantes pour répondre à notre problématique (âge de l'usager, week-end, proximité du point de choc, jour chômé).

A l'issue de cette phase, nous disposons d'un jeu de données de 447136 observations et 98 variables, dont 5 sont des variables continues, les autres étant des variables catégorielles.

En prévision de modélisations impliquant des notions de distance, des procédés de normalisation ont été choisis pour les variables continues : un procédé de Robust Scaling pour la latitude et la longitude, des transformations sinus/cosinus pour l'heure et le mois, une normalisation min/max pour l'âge des usagers.

Une première analyse nous a conduit à identifier les éléments suivants comme facteurs potentiellement importants (à savoir avec une p-valeur à l'issue d'un test d'indépendance du χ^2 inférieure à 5% et un V de Cramer supérieur à 0,1):

- La présence ou non d'équipements de sécurité (ceinture, casque, gants)
- La catégorie d'usagers avec dans l'ordre d'importance identifiée, les motards, les automobilistes, les piétons, les cyclistes, l'implication d'un poids lourds
- La présence d'un obstacle fixe
- La circulation sur route départementale, puis sur voie communale
- Le fait d'être en agglomération
- La collision avec un autre véhicule, ou plusieurs autres véhicules
- La proximité de l'usager avec le point de choc
- Le caractère bidirectionnel de la voie empruntée
- La présence d'une courbe dans le tracé de la route
- L'âge de l'usager
- La circulation de nuit sans éclairage

Quelques facteurs ont été spécifiquement étudiés en fin de ce rapport (analyse temporelle, disparité spatiale, âge et sexe des usagers) pour avoir quelques constats généraux sur leurs liens avec la variable cible, et pour offrir des visualisations intéressantes des tendances observables avec ce jeu de données.

Le jeu de données que nous avons préparé doit dorénavant pouvoir nous servir de base pour l'analyse de notre problème de classification supervisé. Différentes perspectives sont envisagées : l'analyse à l'aide d'arbres de décision pour aller plus loin dans l'identification des facteurs influents, une approche multi-classes de la classification en conservant les 4 classes de gravité (tué, blessé hospitalisé, blessé léger, indemne), et une approche de classification à seulement 2 classes en considérant d'une part les tués et blessés hospitalisés, et d'autre part les blessés légers et usagers indemnes.

L'ensemble des éléments fournis dans ce rapport sont visibles sur le repo GitHub dont l'adresse est : https://github.com/DataScientest-Studio/sept23_cds_accidents2.

I.8 Perspectives de modélisation

Pour la prédiction de la gravité des accidents routiers en France, nous avons choisi de traiter le problème sous deux angles :

- prédire le nombre journalier de chacune des modalités de notre cible ('grav') par le biais de séries temporelles,
- prédire l'état de gravité de la personne accidentée en fonction des caractéristiques de l'accident par une classification multi-classes.

Pour chacune de ces approches, les prévisions sont réalisées par un apprentissage supervisé.

Séries temporelles

Dans le cas des séries temporelles, nous souhaitons prédire le nombre d'indemnés, de blessés légers, de blessés hospitalisés et de tués par jour pour les prochains mois. Pour comparer les différents modèles utilisés, nous employons 3 métriques :

- l'erreur absolue moyenne (MAE : mean absolute error),
- le carré moyen des erreurs (MSE : mean squared error),
- l'erreur quadratique moyenne (RMSE : root mean squared error).

La MAE quantifie les erreurs réalisées par le modèle en pénalisant autant les grandes que les petites erreurs. Étant dans la même unité que la variable à prédire, elle permet d'interpréter facilement l'erreur du modèle.

La MSE quantifie les erreurs réalisées par le modèle, mais contrairement à la MAE, elle pénalise plus fortement les grandes erreurs que les petites erreurs. En revanche, elle ne permet pas d'interpréter facilement le modèle car elle n'est pas dans la même unité que le modèle.

La RMSE est la racine carrée de la MSE. Contrairement à la MSE, elle s'exprime dans la même unité que la variable à prédire permettant donc une interprétation plus facile du modèle.

Classification multi-classes

Dans le cas de la classification multi-classes, nous utilisons aussi 3 métriques :

- l'accuracy,
- le f1-score,
- la matrice de confusion.

L'accuracy (nombre de prédictions correctes / nombre total de prédictions) permet de connaître si le modèle est performant ou pas et s'il y a du sur- ou du sous-apprentissage.

Le choix du f1-score est dû au fait que nous avons un jeu de données très déséquilibré. En effet, le f1-score étant la moyenne harmonique de la précision et du rappel, cela permet d'optimiser la précision et le rappel pour obtenir le maximum de vrais positifs.

Enfin, la matrice de confusion permet de comparer dans le détail les modèles ayant des f1-scores semblables.

II PRÉDICTION DU NOMBRE D'INDEMNES, DE BLESSÉS LÉGERS, DE BLESSÉS HOSPITALISÉS ET DE TUÉS PAR JOUR

II.1 Séries temporelles

II.1.1 Création des jeux de données

Pour les séries temporelles, nous allons créer 4 jeux de données en fonction des 4 modalités de la variable cible ‘grav’ à partir de notre jeu de données initial, soit :

- 1 jeu de données pour la modalité ‘indemnes’,
- 1 jeu de données pour la modalité ‘blessés légers’,
- 1 jeu de données pour la modalité ‘blessés hospitalisés’,
- 1 jeu de données pour la modalité ‘tués’.

Pour réaliser cela, nous procédons en plusieurs étapes :

- conservation uniquement des variables ‘jour’, ‘mois’, ‘an’, ‘grav’,
- séparation en 4 jeux de données, en récupérant les lignes correspondant à la modalité choisie,
- création d’une variable date au format datetime à partir des variables ‘jour’, ‘mois’, ‘an’ et suppression de ces dernières,
- création d’une variable ‘Nbre_Acc’ qui correspond au nombre d’accidents par jour,
- mise en index de la date.

Cependant, le jeu de données initial que nous avons choisi pour traiter le problème de prédiction de la gravité des accidents routiers en France regroupe les données recueillies pour les années 2019 à 2022. Or la pandémie de Covid-19 et les confinements associés, en mars 2020 et, de moindre mesure, en octobre 2020, ont biaisé les données au niveau temporel. En effet, lors des confinements peu de personnes étaient autorisées à sortir, donc le nombre d'accidents a fortement chuté, induisant une diminution du nombre de cas pour chacune des modalités de notre variable cible ‘grav’ (Figure 44).

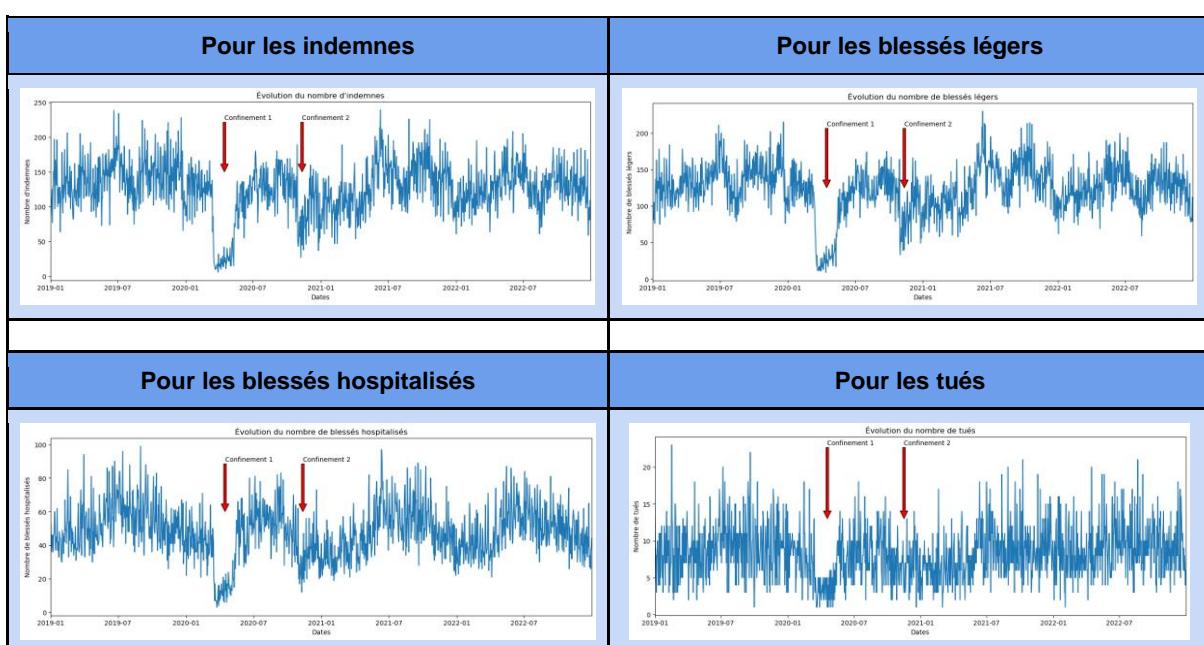


Figure 44 : Évolutions du nombre de cas par jour entre 2019 et 2022

Pour ne pas avoir de données biaisées pour prédire le nombre de cas de chacune des modalités de la variable cible, nous avons décidé de ne prendre que les années 2021 et 2022.

II.1.2 Etude de la saisonnalité

L'étude de la saisonnalité a été effectuée en s'inspirant de la publication suivante : <https://blog.statoscop.fr/timeseries-4.html>.

E. Tendance

Pour visualiser la tendance, on effectue un lissage de la courbe à l'aide des moyennes mobiles sur une fenêtre glissante d'observations. En choisissant différentes tailles de fenêtres glissantes (correspondant à 1, 2, 3 et 4 semaines et à 6 mois et 1 an), on peut clairement observer une tendance linéaire et stable pour une fenêtre de 365 jours quelle que soit la modalité (Figure 45).

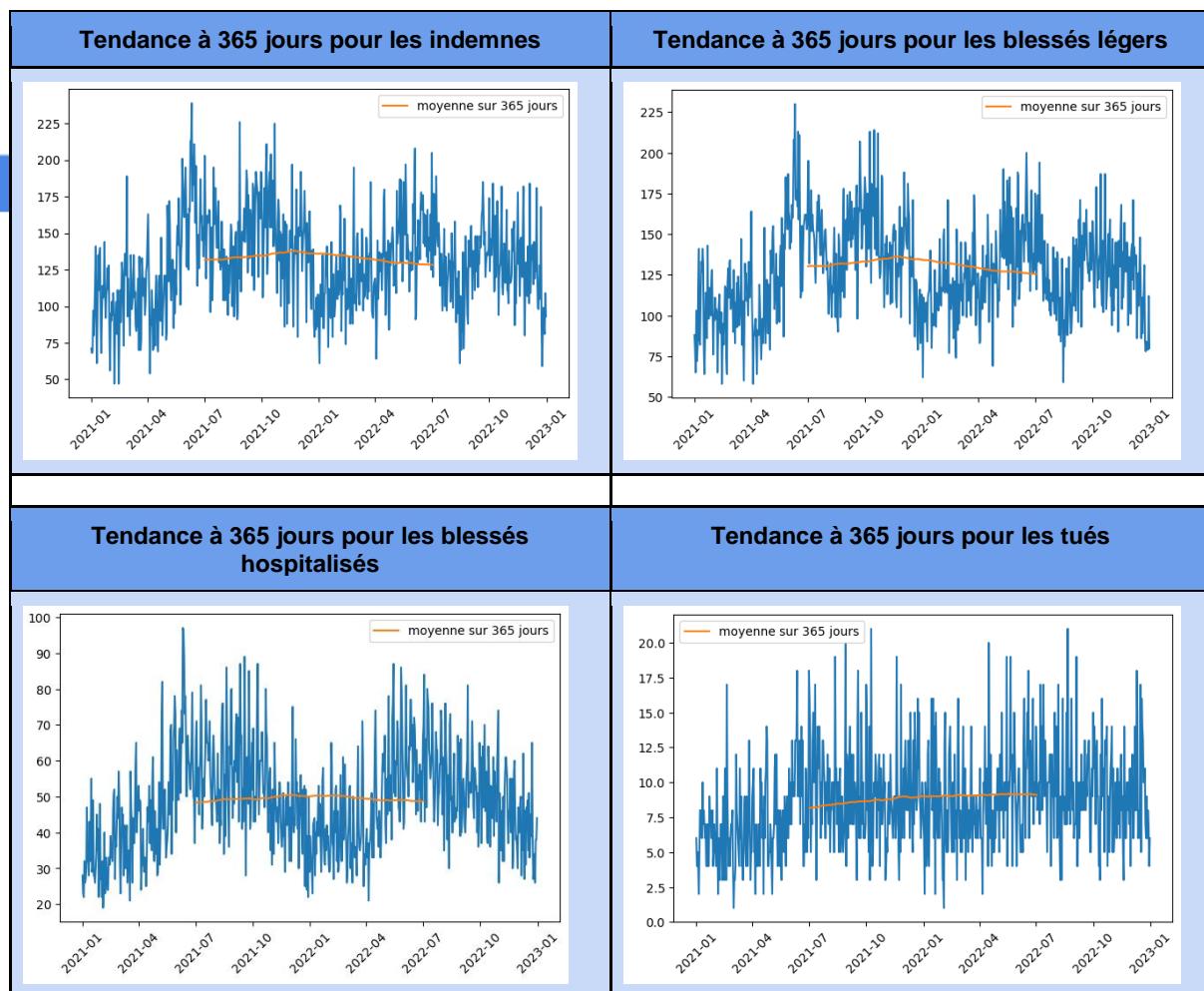


Figure 45 : Lissage de la courbe sur une fenêtre glissante de 365 jours.

F. Saisonnalité

Une saisonnalité annuelle est visible dans la Figure 45, avec des pics vers juin et octobre et des « creux » hivernaux dans les quatre séries (moins visible dans la série des tués, où le nombre de cas est plus faible et le bruit plus important). Par ailleurs, le nombre d'accidentés de chaque classe est aussi dépendant du jour de la semaine, cette saisonnalité hebdomadaire étant bien visible dans la Figure 46.

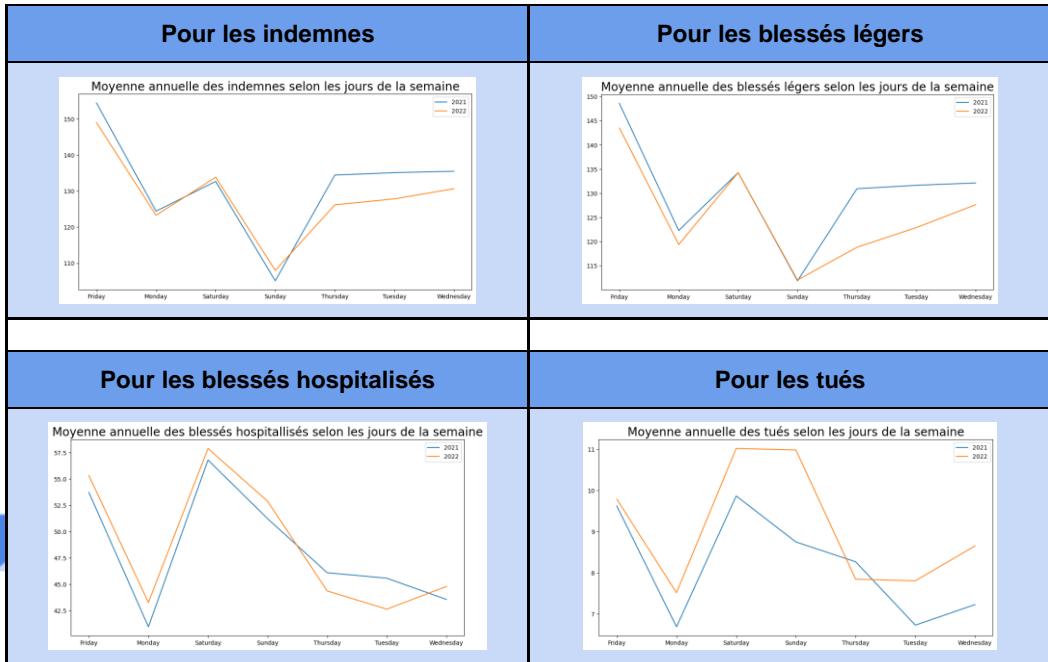


Figure 46 : Moyennes annuelles du nombre de cas par jours pour les années 2021 et 2022

G. Bruit

Comme le modèle est additif, le bruit s'obtient en soustrayant à la série originale la tendance et la saisonnalité (Figure 47).

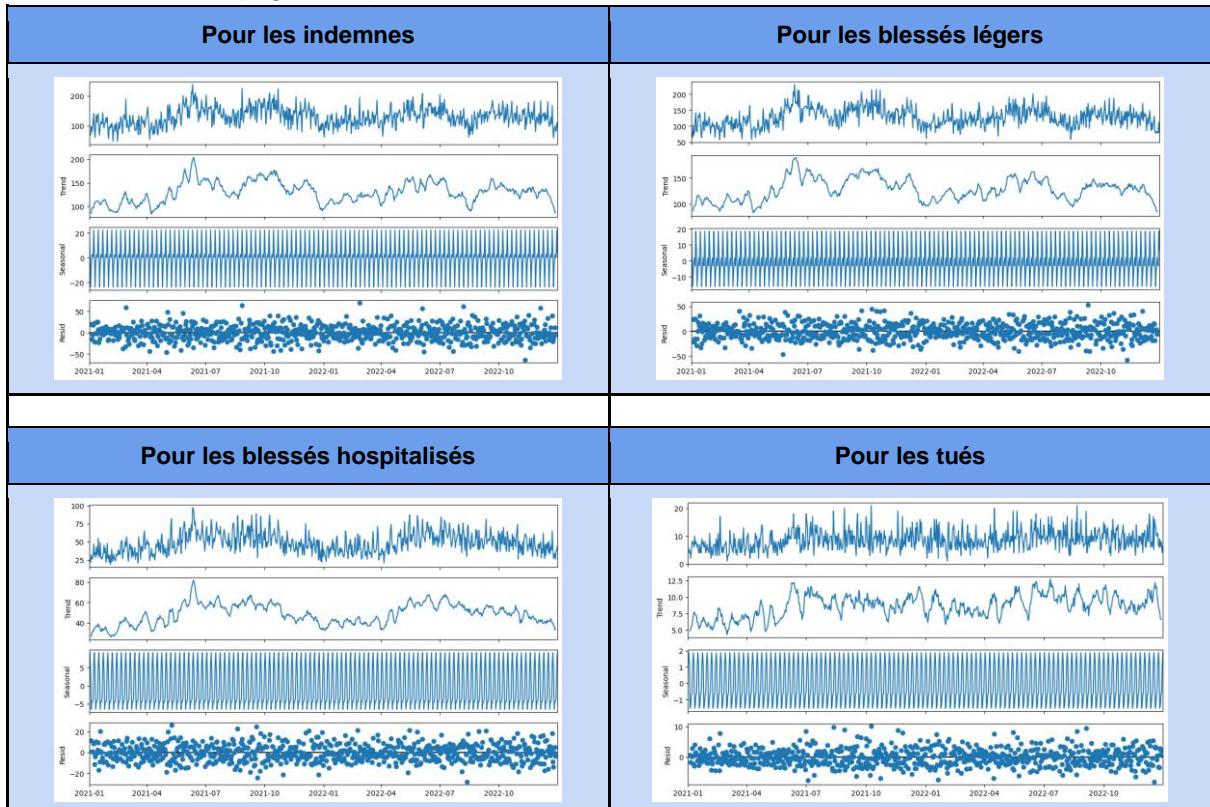


Figure 47 : Décompositions des séries temporelles

II.1.3 Création d'une baseline

A. Méthode naïve avec un shift de 1

On crée une nouvelle colonne correspondant au décalage d'une journée des valeurs de la série. On fait de même pour tous les jeux de données et on évalue les résultats avec les métriques préalablement choisies (MAE, MSE et RMSE) pour obtenir les courbes suivantes (Figure 48) :

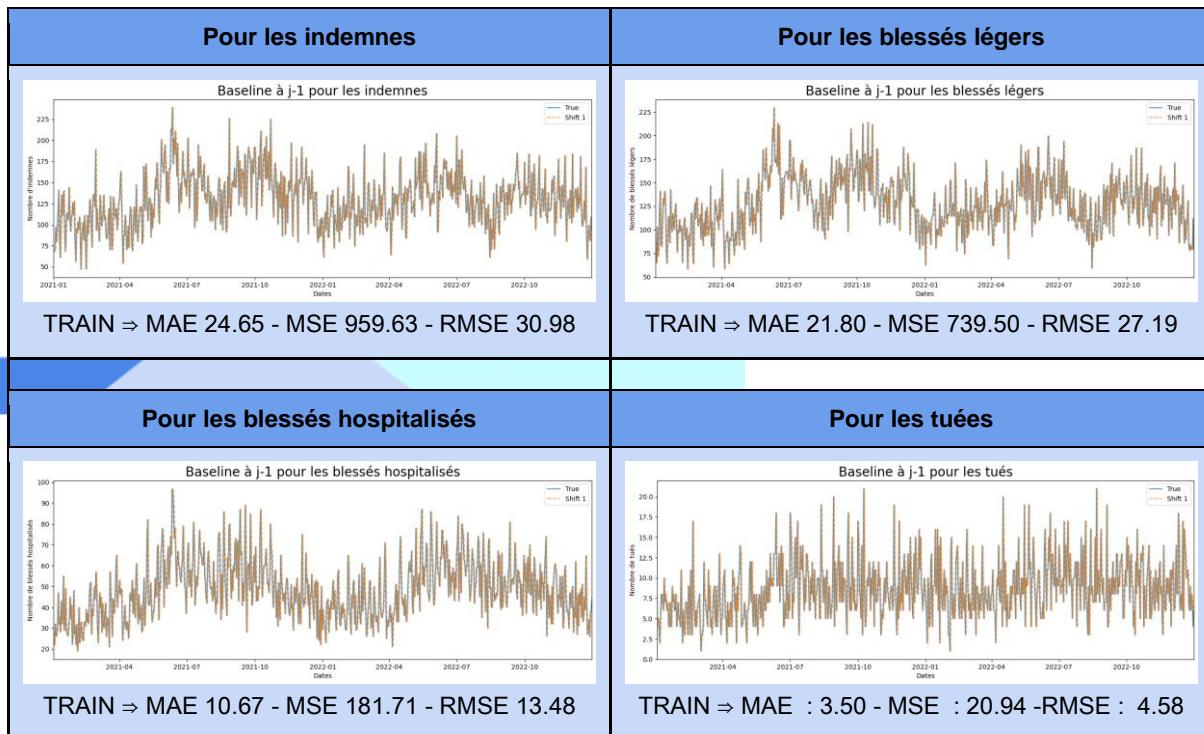


Figure 48: Baselines avec shift de 1 jour

B. Méthode naïve avec une moyenne sur 7 jours et un shift de 1

Comme les jeux de données présentent une saisonnalité hebdomadaire, nous décidons de créer une autre baseline moyennée sur 7 jours. La création de cette nouvelle colonne se fait en prenant les valeurs de la série sur 7 jours que l'on moyenne et que l'on décale d'une journée. En faisant de même avec tous les jeux de données et en calculant les métriques, on obtient les courbes de la Figure 49.

On observe que la moyenne sur 7 jours permet d'obtenir de meilleurs résultats ce qui tend à confirmer notre saisonnalité hebdomadaire.

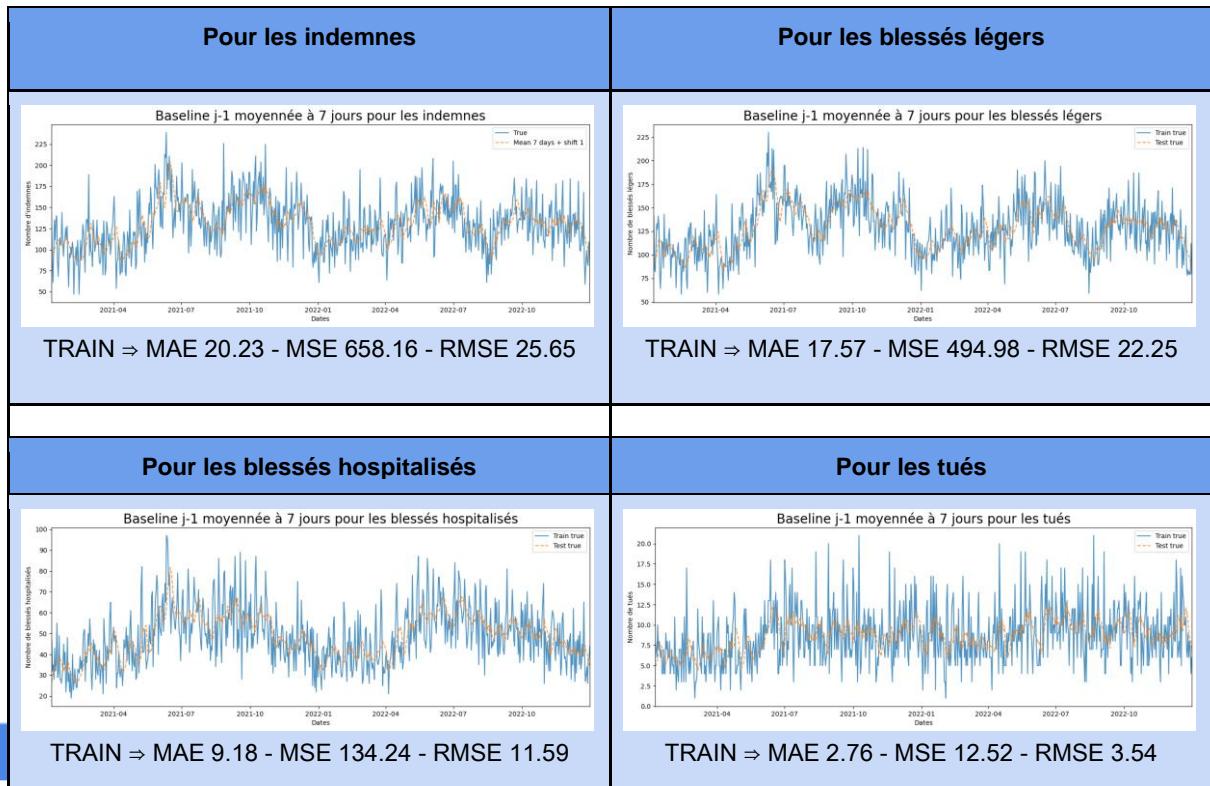


Figure 49 : Baselines moyennées sur 7 jours + shift de 1 jour

C. SARIMAX + Exog(Série de Fourier)

Si la baseline moyennée à 7 jours permet de prendre en compte la saisonnalité hebdomadaire, elle ne prend pas en compte la saisonnalité annuelle. Pour cela, nous décidons d'effectuer une modélisation avec SARIMAX en prenant en variable exogène une série de Fourier. L'implémentation du modèle a été effectuée en se basant sur la méthodologie issue de l'analyse de la qualité de l'air expliquée dans <https://www.kaggle.com/code/saisatishmasina/sarima-with-fourier-terms>.

Avant de modéliser l'algorithme, on effectue :

- une vérification de la stationnarité des jeux de données avec un test ADF (Augmented Dickey-Fuller). La stationnarité est vérifiée pour les tués. Par contre, il faut faire une différenciation d'ordre 1 pour les autres jeux de données pour qu'ils deviennent stationnaires.
- une vérification de la saisonnalité annuelle en traçant le périodogramme et en vérifiant la densité spectrale liée à chaque période. La saisonnalité sur 365 jours est vérifiée pour tous les jeux de données.
- une séparation des jeux de données en train (90%) et test (10%) de manière chronologique.
- une extrapolation de la série de Fourier sur l'ensemble des données qui est ensuite intégrée dans train et test via une nouvelle variable exogène.
- une approximation des paramètres via le traçage des courbes d'autocorrélation et d'autocorrélation partielle. Dans tous les cas, afin d'obtenir un bruit blanc faible, nous rajoutons une différenciation en plus de celle déjà effectuée pour la stationnarisation : ce qui donne une différenciation d'ordre 1 pour les tués et une différenciation d'ordre

2 pour les autres modalités. Les courbes permettent aussi de vérifier la saisonnalité hebdomadaire.

Pour la modélisation avec SARIMAX, nous recherchons les paramètres p , q , P , Q qui minimisent l'AIC (Akaike Information Criterion). Pour cela, nous définissons les paramètres ainsi :

- l'ordre d de la partie non saisonnière est fixé à 1 pour les tués et à 2 pour les autres (comme nous l'avons défini lors du tracé des courbes d'autocorrélation et d'autocorrélation partielle),
- l'ordre D de la partie saisonnière est fixé à 1 pour prendre en compte la saisonnalité sur 7 jours,
- les ordres p , q , P et Q sont fixés entre 0 et 2 afin de trouver les meilleures valeurs.

De plus, on ajoute la variable exogène de train afin de prendre en compte la saisonnalité sur 365 jours.

Si les résultats obtenus ont des $P>|z|$ supérieurs à 0.5, on modifie alors les meilleurs paramètres afin d'obtenir tous les $P|z|$ inférieurs à 0.5. Les meilleurs modèles pour chaque modalité sont présentés sur la Figure 50.

Dans tous les cas, nous obtenons des modèles avec un bruit blanc et une hétéroscédasticité. Seuls les modèles pour les indemnités et les tués ne présentent pas de distribution normale (la distribution est très légèrement décalée à gauche).

Enfin, nous évaluons le modèle à la fois sur la partie train et sur la partie test (Figure 51). obtiennent des moins bons résultats que la baseline moyennée sur 7 jours avec un shift de 1. De plus ces modèles nécessitent de connaître les valeurs pour faire une extrapolation de la série de Fourier et ne permettent donc pas de prédire les valeurs futures.

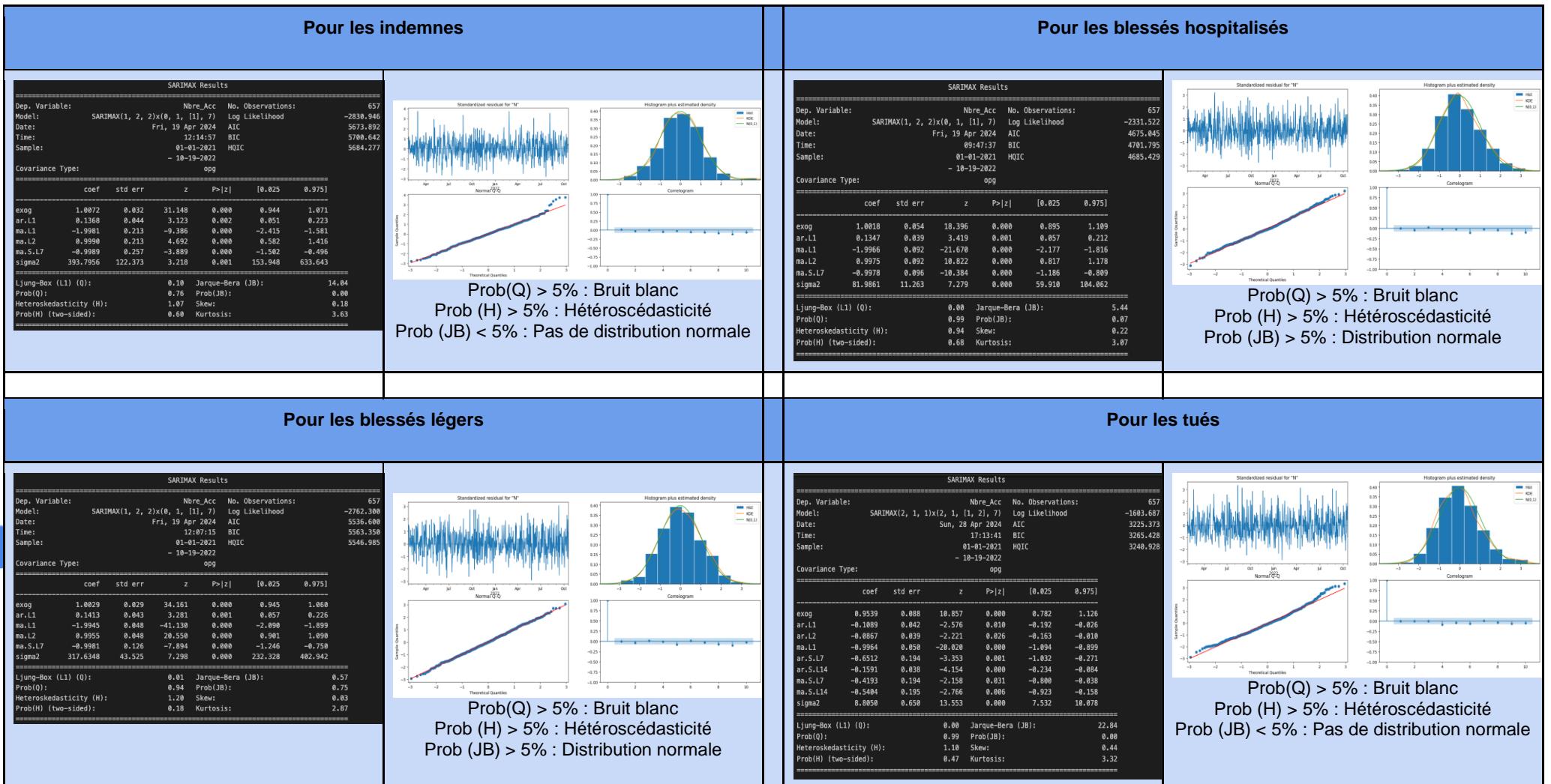


Figure 50 : Résultats de SARIMAX avec en variable exogène une série de Fourier

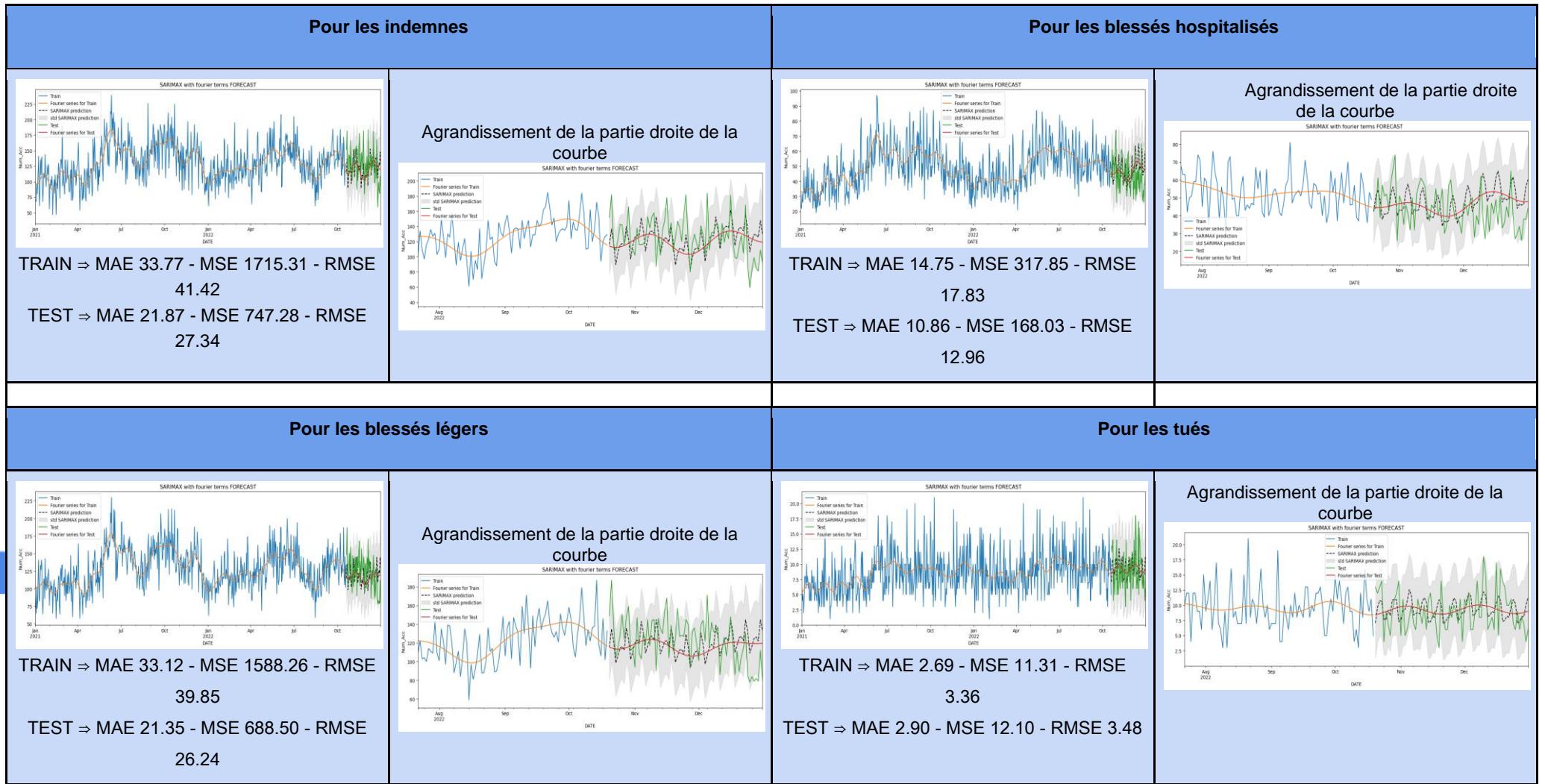


Figure 51 : Évaluations de SARIMAX avec en variable exogène une série de Fourier sur train et test

II.1.4 Prédictions

Pour prédire le nombre d'indemnités, blessés légers, blessés hospitalisés et tués des 6 prochains mois (les 6 premiers mois de 2023), nous utilisons 3 algorithmes :

- 2 algorithmes de machine learning : MSTL (Multiple Seasonal-Trend decomposition using LOESS) et PROPHET,
- 1 algorithme de deep learning : LSTM (Long Short Term Memory).

A. Préparation des jeux de données pour MSTL et PROPHET

Pour implémenter ces algorithmes, il est nécessaire que les jeux de données soient d'une forme particulière. Pour cela, nous transformons nos jeux de données (exemple des premières lignes pour les indemnités) :

- création de la colonne 'ds' à partir de l'index
- création de la colonne 'y' par renommage de la colonne 'Nbre_Acc'
- création de la colonne 'unique_id' en lui donnant la valeur 1 pour toutes les lignes

	Nbre_Acc			
2021-01-01	71	ds	y	unique_id
2021-01-02	68	0	2021-01-01	71

⇒

	ds	y	unique_id
0	2021-01-01	71	1
1	2021-01-02	68	1

Puis on sépare le jeu de données en train (90%) et test (10%) de manière chronologique.

B. MSTL

Les modèles MSTL ont été implémentés à l'aide de la documentation de NIXTLA : <https://nixtlaVerse.nixtla.io/statsforecast/docs/models/multipleseasonaltrend.html/>.

Dans les modèles MSTL, nous définissons 2 paramètres :

- season_length qui est défini à [7, 7 * 52] pour la saisonnalité hebdomadaire et la saisonnalité annuelle,
- trend_forcaster où nous essayons successivement 3 types de tendance (AutoARIMA, AutoTheta et AutoCES).

De plus, nous mettons une fréquence 'D' pour l'implémentation du StatsForecast pour avoir une fréquence des jours calendaires.

Exemple d'implémentation avec un trend_forcaster basé sur AutoARIMA :

```
models = [MSTL(season_length=[7, 7 * 52],  
| | | | | trend_forcaster=AutoARIMA(prediction_intervals=ConformalIntervals(n_windows=3, h=horizon)))]  
sf = StatsForecast(models = models,  
| | | | | freq = 'D',  
| | | | | n_jobs = -1)
```

Les résultats obtenus pour chacun de ces modèles sont les suivants (Figure 52). Dans tous les cas, un surapprentissage sur la partie train tend à apparaître

Pour les indemnées

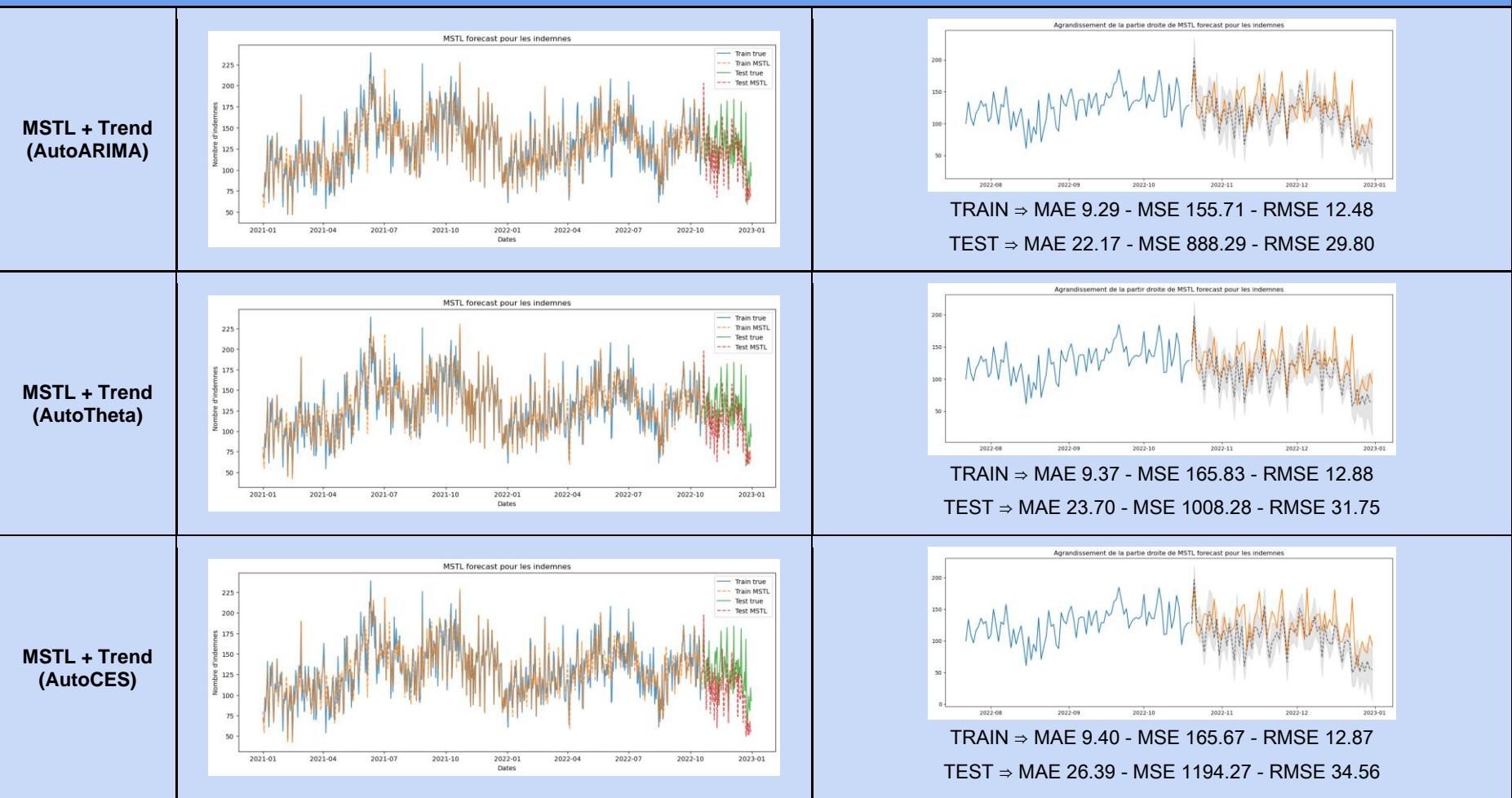


Figure 52 : Évaluation de MSTL avec différents trend_forecaster sur train et test

Pour les blessés légers

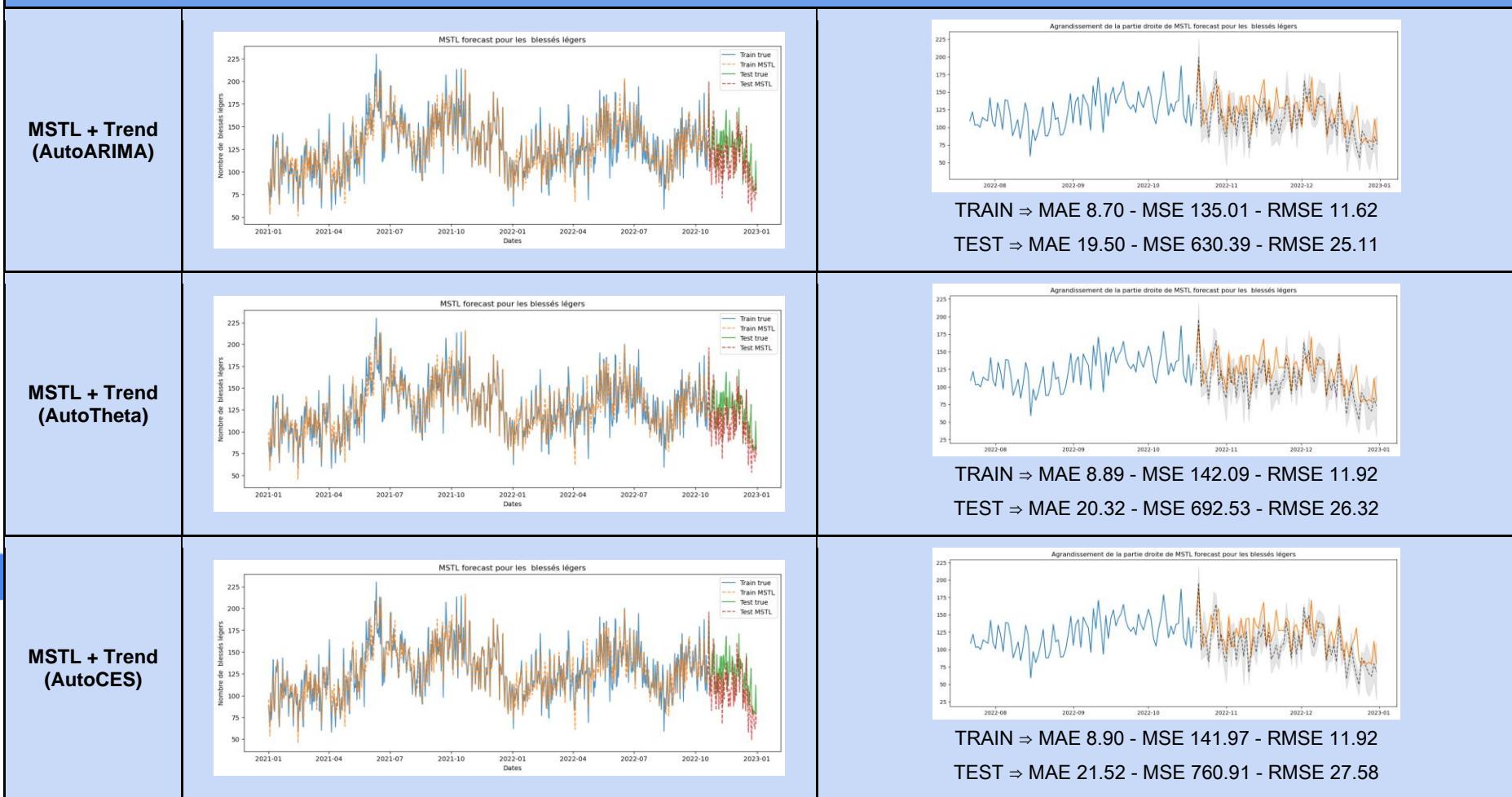


Figure 52 : Évaluation de MSTL avec différents trend_forecaster sur train et test

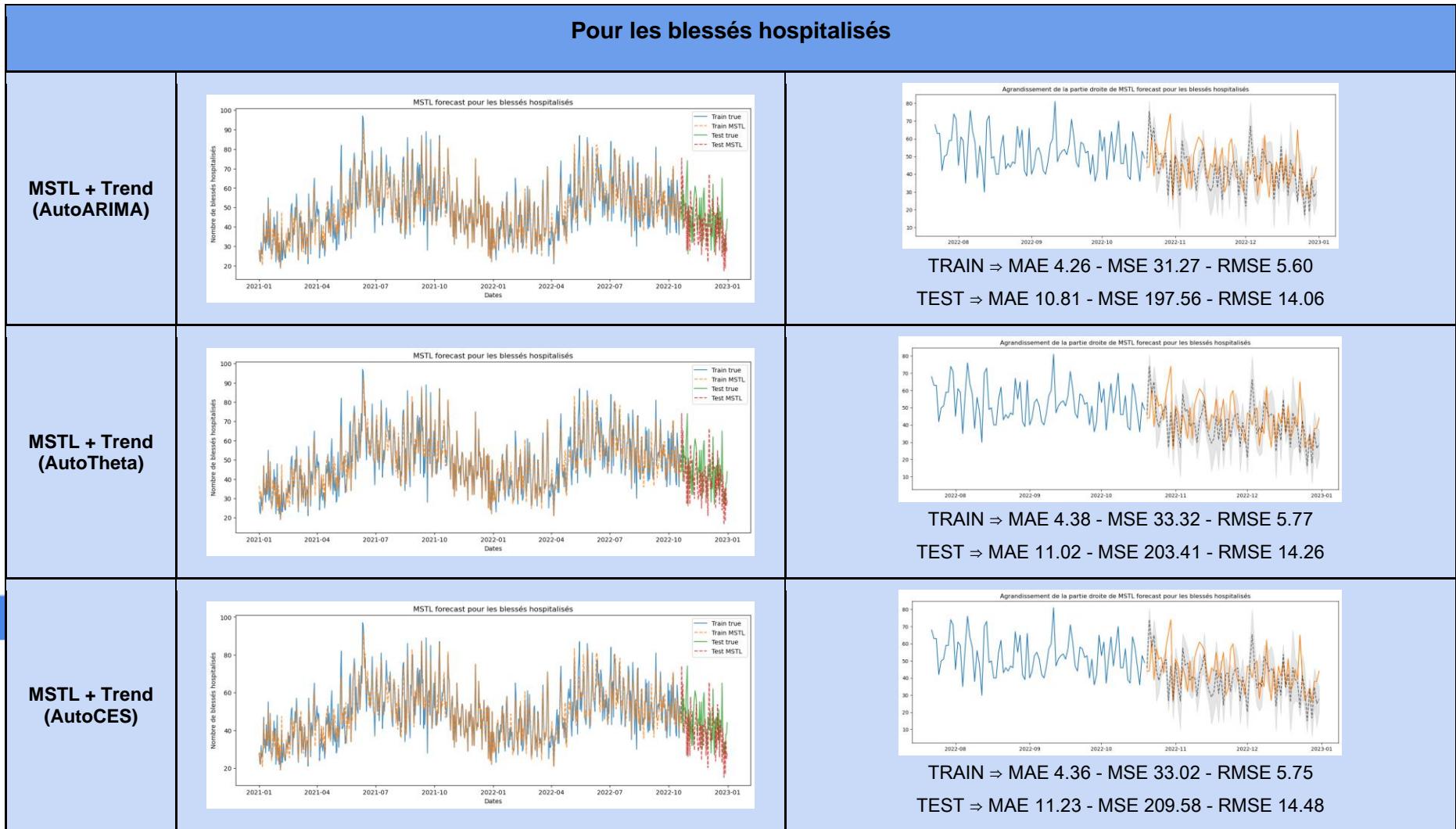


Figure 52 : Évaluation de MSTL avec différents trend_forecaster sur train et test

Pour les tués

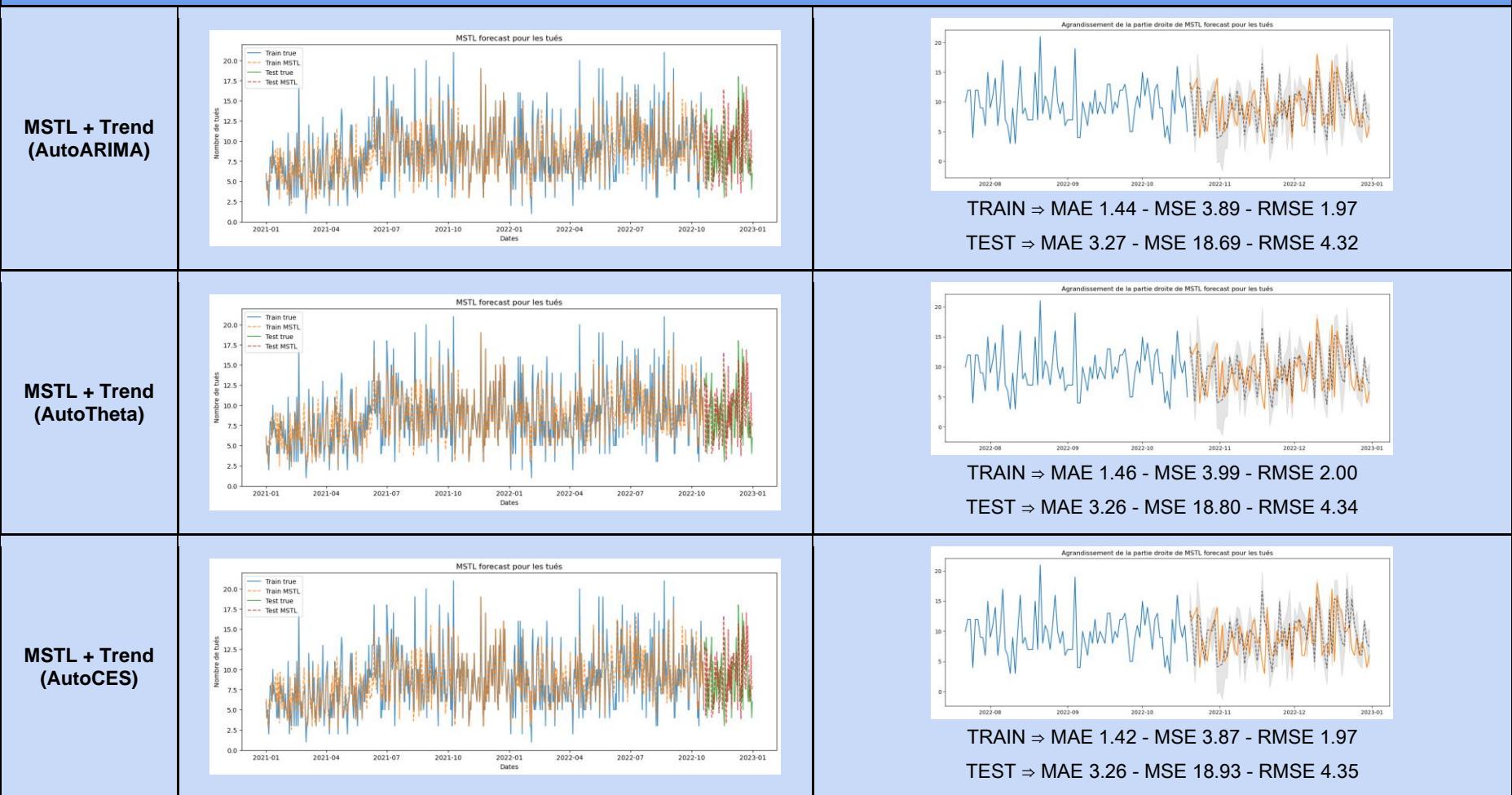


Figure 52: Évaluation de MSTL avec différents trend_forecaster sur train et test

C. PROPHET

L'implémentation de cet algorithme a été réalisée à l'aide de la documentation de PROPHET : <https://facebook.github.io/prophet/>.

Pour cet algorithme, nous avons fait le choix de l'utiliser de 3 manières :

- avec les paramètres `yearly_seasonality` et `weekly_seasonality` sur `True` pour indiquer la saisonnalité annuelle et hebdomadaire,
- avec ajout des vacances scolaires communes à la France métropolitaine et aux DOM-TOM,
- avec ajout des jours fériés communs à la France métropolitaine et aux DOM-TOM.

Pour ajouter les vacances scolaires ou les jours fériés, il est nécessaire de créer un nouveau jeu de données de la forme suivante (exemple des premières lignes pour les vacances scolaires) :

	holiday	ds
0	vacances	2021-01-01
1	vacances	2021-01-02

Exemple d'implémentation avec les vacances scolaires :

```
my_model = Prophet(seasonality_mode = 'additive',
                    yearly_seasonality = True,
                    weekly_seasonality = True,
                    daily_seasonality = False,
                    holidays = vacances)
```

Les résultats obtenus pour chacun de ces modèles sont présentés à la Figure 53.

Pour les indemnies

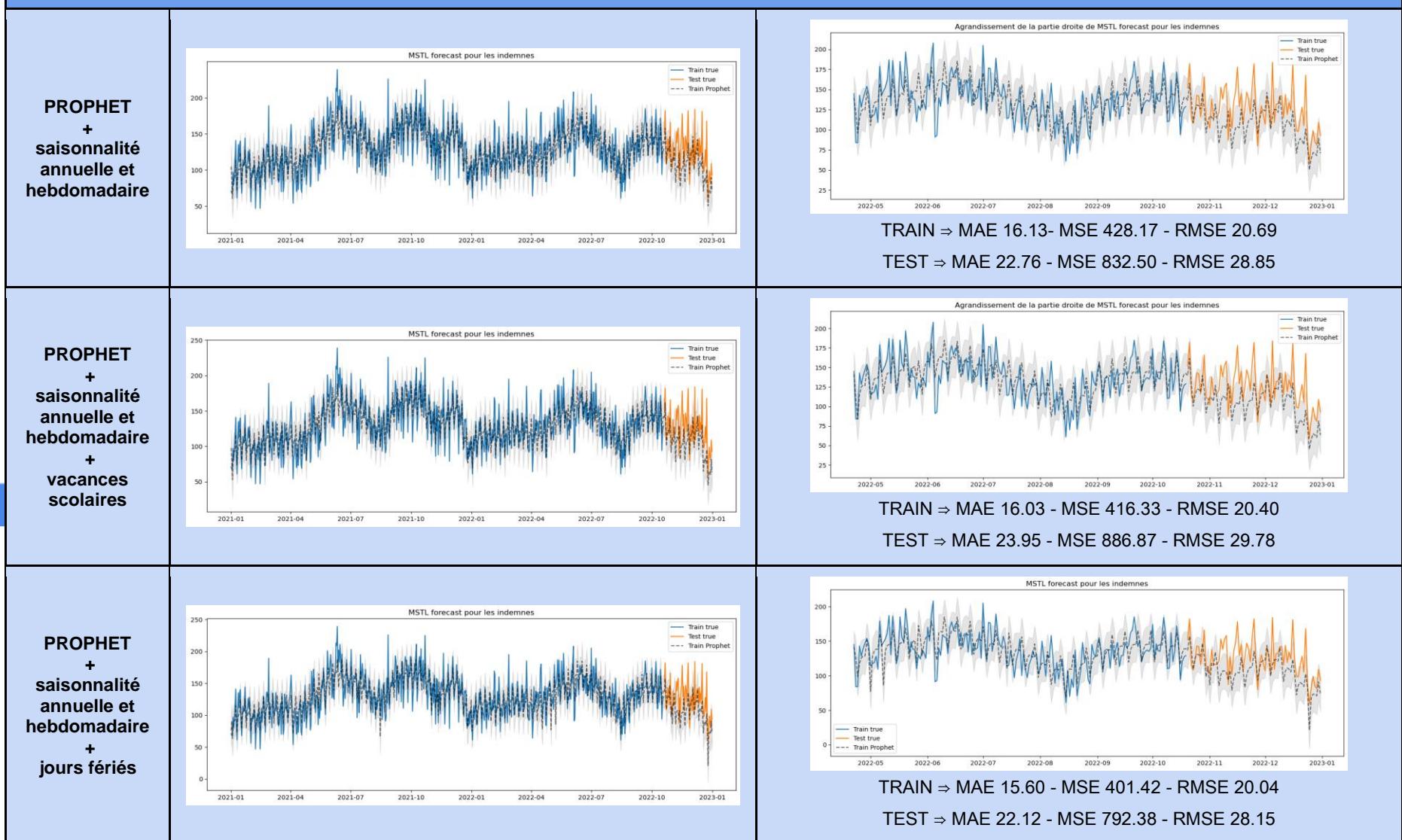


Figure 53 : Évaluation de PROPHET avec différentes saisons sur train et test

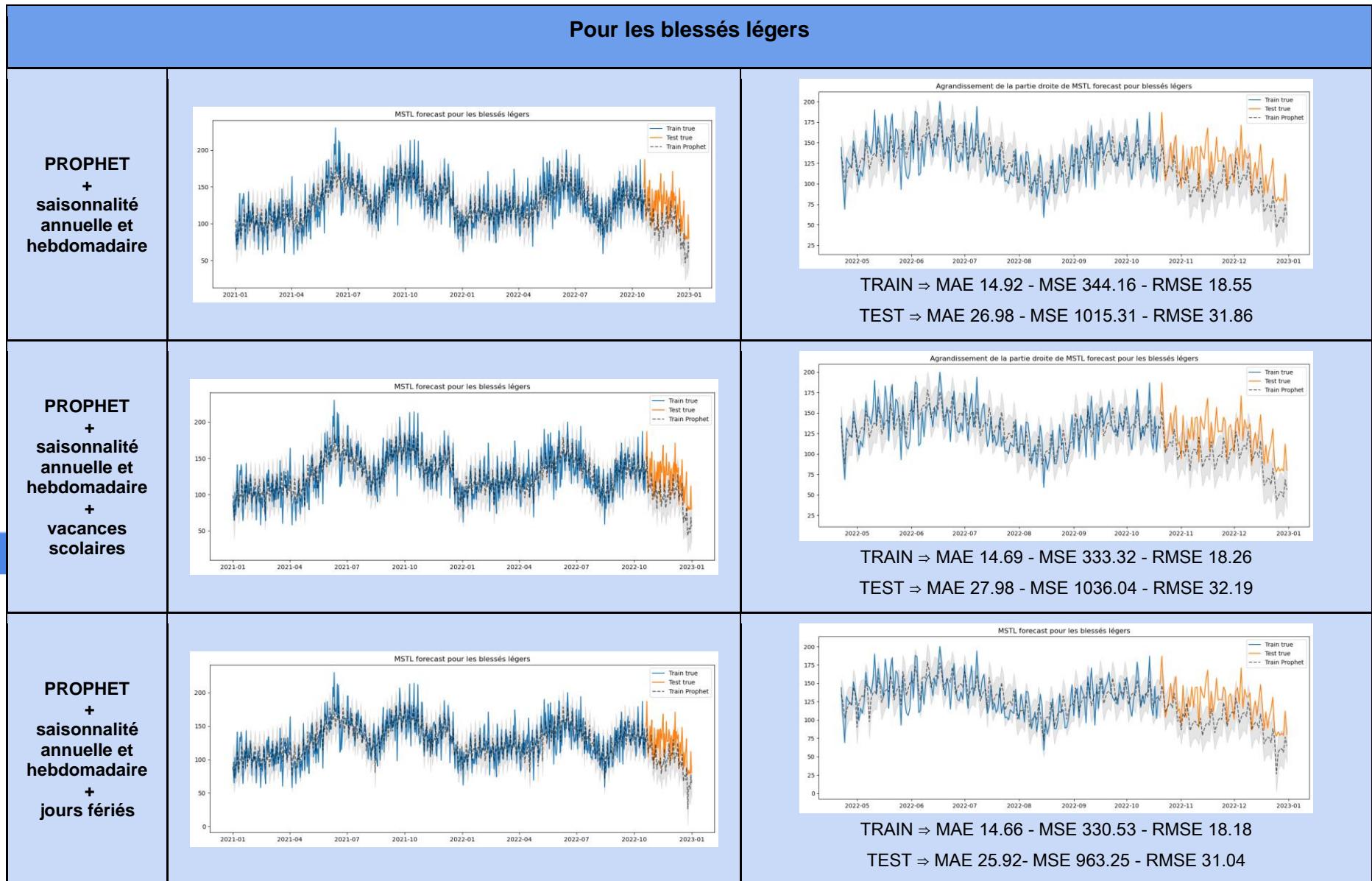


Figure 53 : Évaluation de PROPHET avec différentes saisons sur train et test

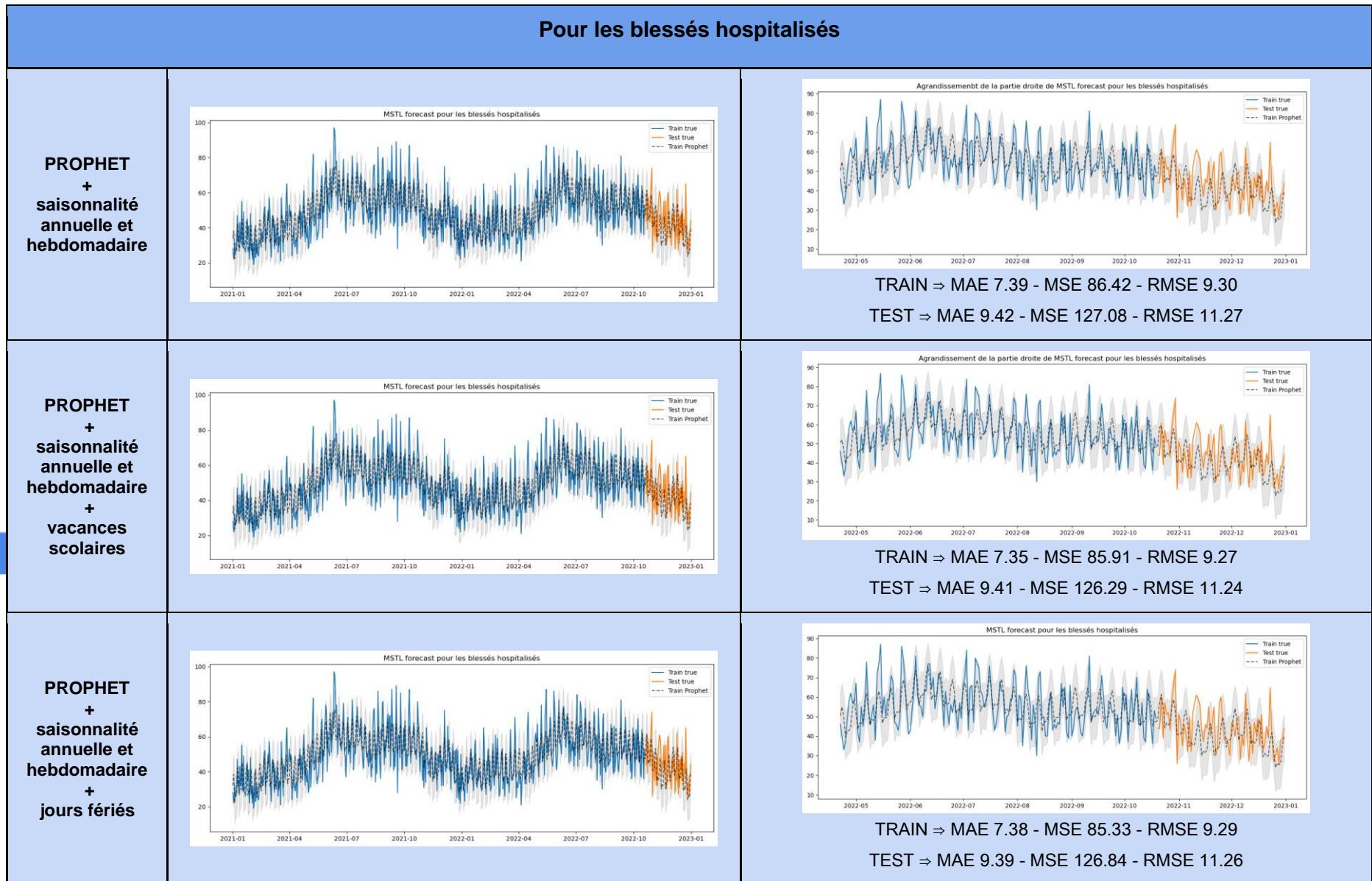


Figure 53 : Évaluation de PROPHET avec différentes saisons sur train et test

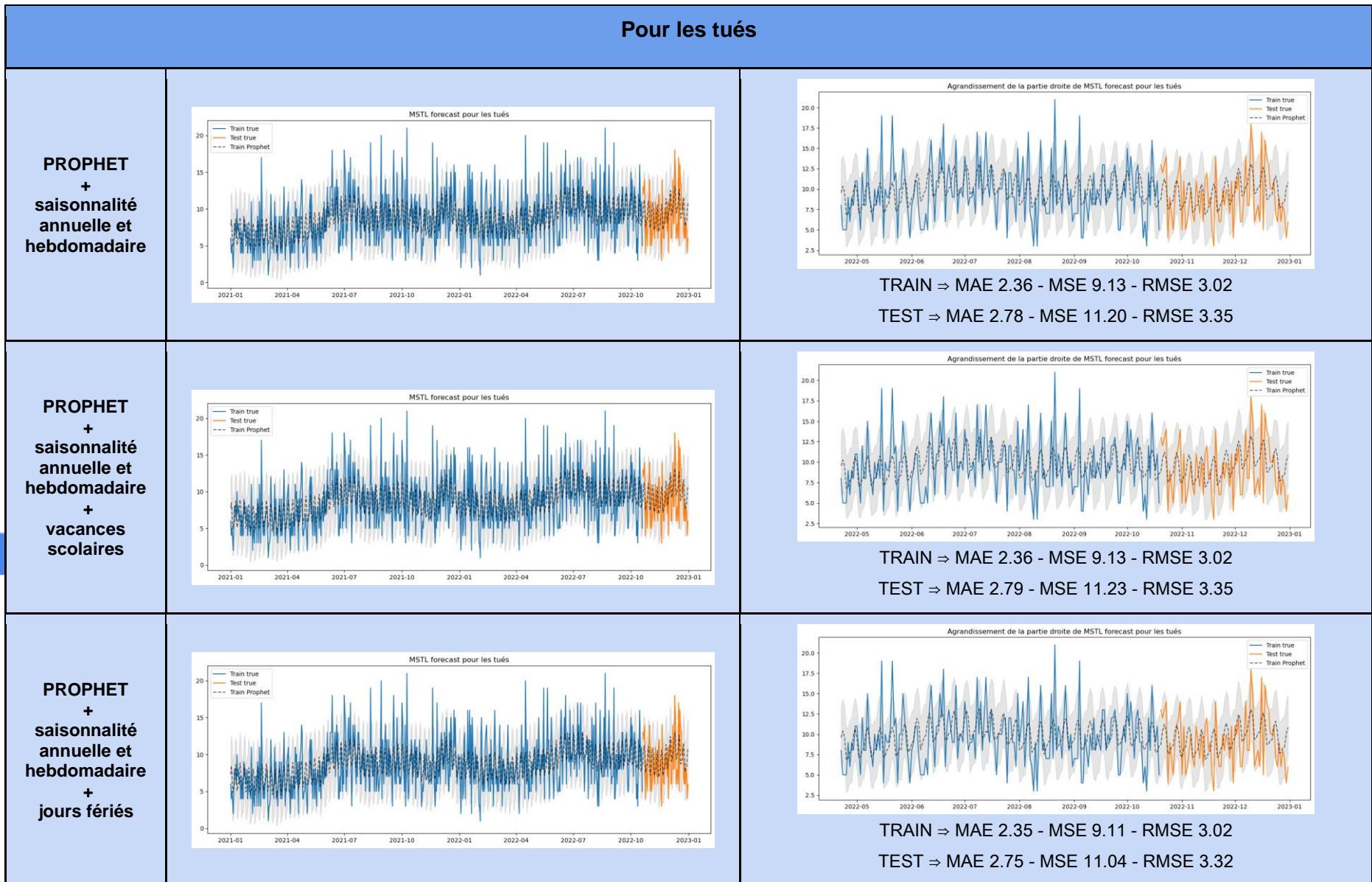


Figure 53 : Évaluation de PROPHET avec différentes saisons sur train et test

D. LSTM

L'implémentation du modèle a été réalisée à l'aide de la publication : <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>.

Pour le deep learning, il est nécessaire de modifier notre jeu de données de la manière suivante

- création d'un jeu de données composé d'un array numpy contenant les valeurs de la colonne 'Nbre_Acc',
- normalisation des données avec MinMaxScaler,
- séparation en train_scaled et test_scaled,
- conversion de train_scaled et de test_scaled en une matrice de séquences de données (avec le look_back qui correspond au nombre de pas de temps précédents à utiliser comme variables d'entrée pour prédire la période suivante),
- remodelage de train_scaled et de test_scaled pour qu'il ait un format [samples, time steps, features]

Nous choisissons d'implémenter un Vanilla MSLT ne comportant qu'une couche LSTM comportant 4 neurones et une couche Dense. Le modèle est compilé avec comme paramètres : loss = 'mean_squared_error' et optimizer = 'adam'. Le modèle est ensuite entraîné sur 100 epochs sur chacun des jeux de données .

```
model = tf.keras.Sequential([
    tf.keras.layers.LSTM(4),
    tf.keras.layers.Dense(1)
])
model.compile(loss='mean_squared_error', optimizer='adam')
history = model.fit(trainX_scaled, trainY_scaled, epochs=100, batch_size=1, verbose=2)
```

Le modèle est ensuite évalué sur les parties train et test. Les résultats obtenus pour un look_back = 31 (soit 31 jours) sont présentés sur la Figure 54.

En augmentant le look_back à 2 mois, les résultats des métriques sont moins bons. Le modèle LSTM est donc un bon modèle pour des prédictions à courte échéance pour nos jeux de données.

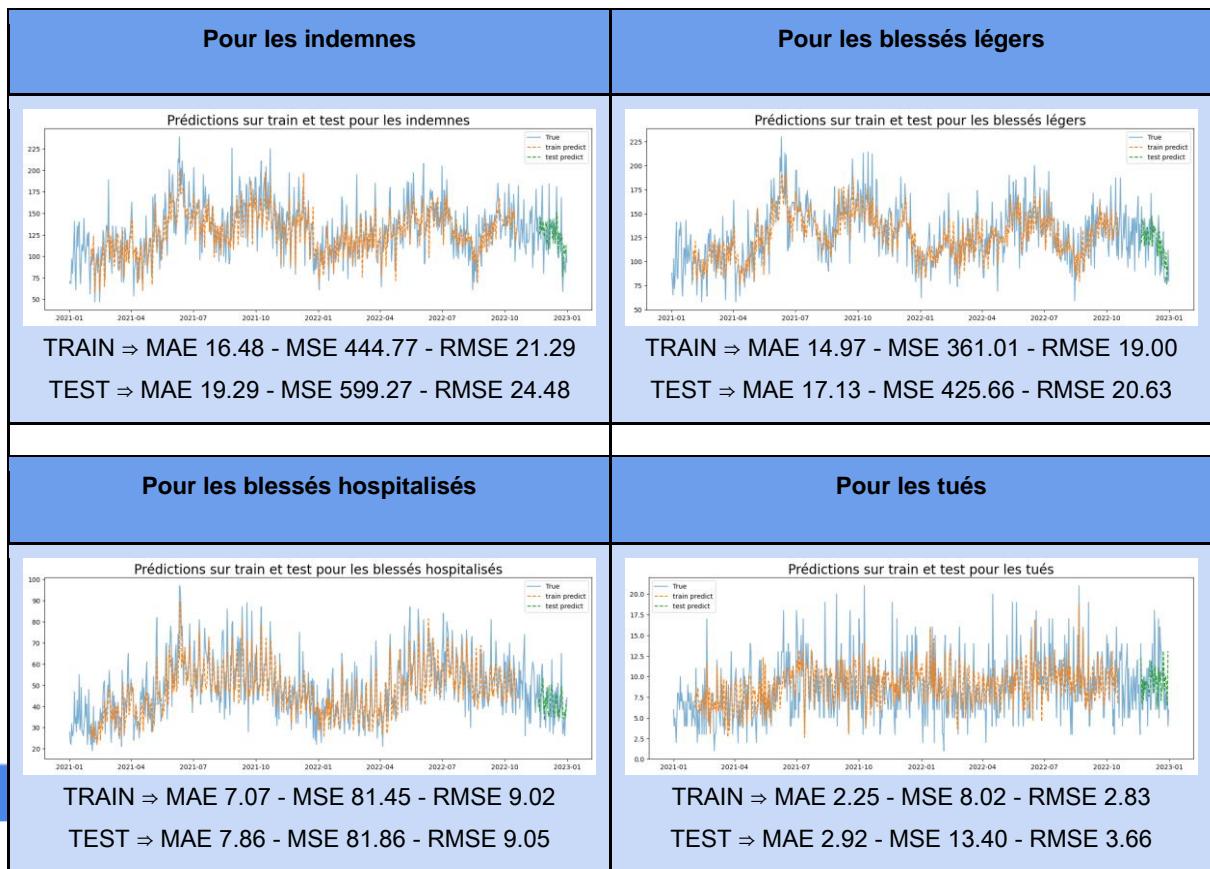


Figure 54 : Évaluations de LSTM sur train et test

E. Choix du meilleur modèle et prévision pour les futures dates

En comparant les résultats obtenus pour les différents algorithmes, nous sélectionnons celui qui minimise la MAE et la RMSE.

Le modèle LSTM ne permet de prédire qu'à courte échéance. En effet, les prévisions se font de manière itérative : prédire une nouvelle valeur, l'ajouter aux valeurs précédentes puis prédire la valeur suivante, et ainsi de suite. Donc plus on s'éloigne des valeurs initiales, plus les résultats risquent d'être biaisés. Nous avons choisi de prédire les valeurs pour le prochain mois avec MSTL afin de visualiser la tendance sur les courbes, mais il ne faudra certainement utiliser que les premières valeurs.

Nous avons retenu les modèles suivants selon la modalité de ‘grav’ (Figure 55) :

- Pour les **indemnées**, les meilleurs résultats sont obtenus pour **LSTM (avec look_back de 31 jours)**. Or celui-ci ne permet de prédire correctement que des données pour les 31 jours suivants. Donc pour une prédiction à plus long terme, il faudrait utiliser **PROPHET avec ajout des jours fériés**.
- Pour les **blessés légers**, les meilleurs résultats sont obtenus pour **LSTM (avec look_back de 31 jours)**. Or celui-ci ne permet de prédire correctement que des données pour les 31 jours suivants. Donc pour une prédiction à plus long terme, il faudrait utiliser **MSTL avec comme trend_forcaster AutoARIMA**.
- Pour les **blessés hospitalisés**, les meilleurs résultats sont obtenus pour **LSTM (avec look_back de 31 jours)**. Or celui-ci ne permet de prédire correctement que des

données pour les 31 jours suivants. Donc pour une prédition à plus long terme, il faudrait utiliser **PROPHET avec ajout des jours fériés**.

- Pour les **tués**, les meilleurs résultats sont obtenus pour **PROPHET avec ajout des jours fériés**.

II.1.5 Conclusion

Pour nos jeux de données (en dehors des tués), les prédictions sont nettement améliorées avec un modèle de deep learning simple. Il pourrait encore être optimisé avec des modèles plus complexes comme Stacked LSTM, Bidirectional LSTM, CNN LSTM, ConvLSTM.

Cependant, même si les résultats sont meilleurs, cela ne permet de prédire qu'à court, voire très court terme. Il est donc nécessaire d'avoir d'autres modèles de machine learning pour faire des prédictions à plus longue échéance.

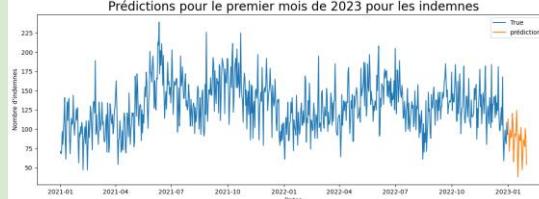
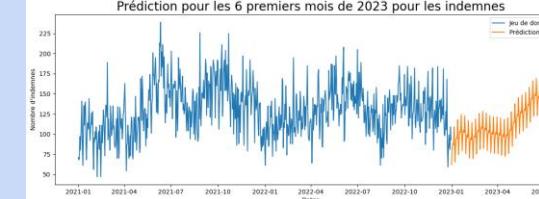
Pour les indemnées						
	MAE	MSE	RMSE			
Baseline (shift de 1)	24.65	959.63	30.98			
Baseline (moyenne sur 7 jours + shift de 1)	20.23	658.16	25.65			
	Train			Test		
	MAE	MSE	RMSE	MAE	MSE	RMSE
SARIMAX + exog(Fourier)	33.77	1715.31	41.42	21.87	747.28	27.34
MSTL + AutoARIMA	9.29	155.71	12.48	22.17	888.29	29.80
MSTL + AutoTheta	9.37	165.83	12.88	23.70	1008.28	31.75
MSTL + AutoCES	9.40	165.67	12.87	26.39	1194.27	34.56
PROPHET	16.13	428.17	20.69	22.76	832.50	28.85
PROPHET + vacances scolaires	16.03	416.33	20.40	23.95	886.87	29.78
PROPHET + jours fériés	15.60	401.42	20.04	22.12	792.38	28.15
LSTM (look_back de 31 jours)	16.47	444.77	21.09	19.29	599.27	24.48
Prévisions à 1 mois avec LSTM (look_back de 31 jours) Prédictions pour le premier mois de 2023 pour les indemnées				Prévisions à 6 mois avec PROPHET + jours fériés Prédiction pour les 6 premiers mois de 2023 pour les indemnées		
						

Figure 55 : Comparaison des modèles et prédictions à 1 et 6 mois pour les indemnées

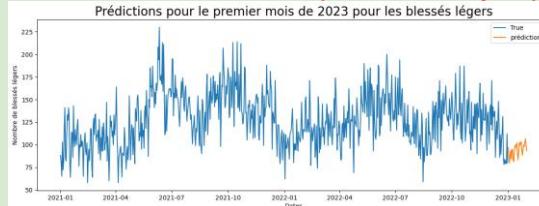
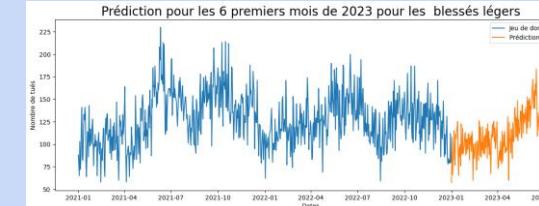
Pour les blessés légers						
	MAE	MSE	RMSE			
Baseline (shift de 1)	21.80	739.50	27.19			
Baseline (moyenne sur 7 jours + shift de 1)	17.57	494.98	22.25			
	Train			Test		
	MAE	MSE	RMSE	MAE	MSE	RMSE
SARIMAX + exog(Fourier)	33.12	317.85	17.83	21.35	688.50	26.24
MSTL + AutiARIMA	8.70	135.01	11.62	19.50	630.39	25.11
MSTL + AutoTheta	8.89	142.09	11.92	20.32	692.53	26.32
MSTL + AutoCES	8.90	141.97	11.92	21.52	760.91	27.58
PROPHET	14.92	344.16	18.55	26.98	1015.31	31.86
PROPHET + vacances scolaires	14.69	333.32	18.26	27.98	1036.04	32.19
PROPHET + jours fériés	14.66	330.53	18.18	25.92	963.25	31.04
LSTM (look_back de 31 jours)	14.97	361.01	19.00	17.13	425.66	20.63
	Prévisions à 1 mois avec LSTM (look_back de 31 jours)			Prévisions à 6 mois avec MSTL + AutiARIMA		
	Prédictions pour le premier mois de 2023 pour les blessés légers 			Prédition pour les 6 premiers mois de 2023 pour les blessés légers 		

Figure 56 : Comparaison des modèles et prédictions à 1 et 6 mois pour les blessés légers

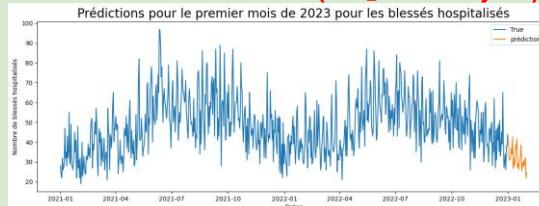
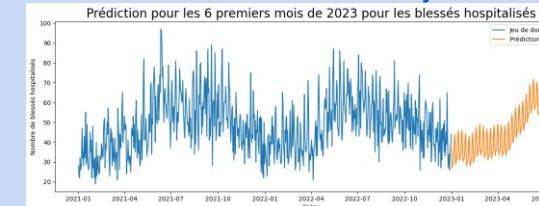
Pour les blessés hospitalisés						
	MAE	MSE	RMSE			
Baseline (shift de 1)	10.67	181.71	13.48			
Baseline (moyenne sur 7 jours + shift de 1)	9.18	134.24	11.59			
	Train			Test		
	MAE	MSE	RMSE	MAE	MSE	RMSE
SARIMAX + exog(Fourier)	14.75	317.85	17.83	10.86	168.03	12.96
MSTL + AutoARIMA	4.26	31.27	5.60	10.81	197.56	14.06
MSTL + AutoTheta	4.38	33.32	5.77	11.02	203.41	14.26
MSTL + AutoCES	4.36	33.02	5.75	11.23	209.58	14.48
PROPHET	7.39	86.42	9.30	9.42	127.08	11.27
PROPHET + vacances scolaires	7.35	85.91	9.27	9.41	126.29	11.24
PROPHET + jours fériés	7.38	85.33	9.29	9.39	126.84	11.26
LSTM (look_back de 31 jours)	7.07	81.45	9.02	7.86	81.86	9.05
Prévisions à 1 mois avec LSTM (look_back de 31 jours) Prédictions pour le premier mois de 2023 pour les blessés hospitalisés						
						
Prévisions à 6 mois avec PROPHET + jours fériés Prédition pour les 6 premiers mois de 2023 pour les blessés hospitalisés						
						

Figure 57 : Comparaison des modèles et prédictions à 1 et 6 mois pour les blessés hospitalisés

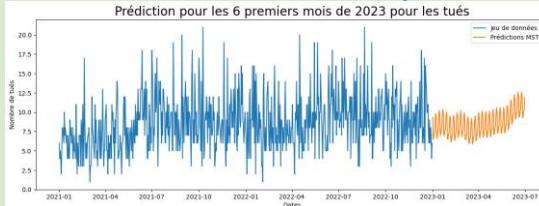
Pour les tués						
	MAE	MSE	RMSE			
Baseline (shift de 1)	3.50	20.94	4.58			
Baseline (moyenne sur 7 jours + shift de 1)	2.76	12.52	3.54			
	Train			Test		
	MAE	MSE	RMSE	MAE	MSE	RMSE
SARIMAX + exog(Fourier)	2.69	11.31	3.36	2.90	12.10	3.48
MSTL + AutoARIMA	1.44	3.89	1.97	3.27	18.69	4.32
MSTL + AutoTheta	1.46	3.99	2.00	3.26	18.80	4.34
MSTL + AutoCES	1.42	3.87	1.97	3.26	18.93	4.35
PROPHET	2.36	9.13	3.02	2.78	11.20	3.35
PROPHET + vacances scolaires	2.36	9.13	3.02	2.79	11.23	3.35
PROPHET + jours fériés	2.35	9.11	3.02	2.75	11.04	3.32
LSTM (look_back de 31 jours)	2.25	8.02	2.83	2.93	13.40	3.66
Prévisions à 6 mois avec PROPHET + jours fériés Prévision pour les 6 premiers mois de 2023 pour les tués						
						

Figure 58 : Comparaison des modèles et prédictions à 1 et 6 mois pour les tués

III PRÉDICTION DE LA GRAVITÉ DE L'ACCIDENT – MACHINE LEARNING

III.1 Régression logistique

III.1.1 Modèle de référence

A. Description du modèle

Le modèle est construit sous forme de **pipeline** en 2 étapes :

- Etape 1 : preprocesser de type **ColumnTransformer** où sont appliquées les étapes de preprocessing décidées à l'analyse préliminaire des données (section I.5) :
 - o Transformation des heures et des mois
 - o Transformation des latitudes et longitudes (RobustScaler)
 - o Transformation de l'âge (MinMaxScaler)
 - o Encodage des variables catégorielles (one_hot_encoder)
 - o Passage des autres variables de la base de données (remainder)
- Etape 2 : modèle de régression logistique

B. Paramètres du modèle de référence

penalty	'l2'	intercept_scaling	1	multi_class	'ovr'
dual	False	class_weight	'balanced'	verbose	0
tol	0.0001	random_state	None	warm_start	False
C	1.0	solver	'lbfgs'	n_jobs	None
fit_intercept	True	max_iter	5000	l1_ratio	None

Tableau 4 : Paramètres du modèle de régression logistique pris comme référence

Une validation croisée sur 3 échantillons conduit à un score de :

- 0.6007 ± 0.0007 sur l'échantillon d'apprentissage,
- et de 0.5990 ± 0.0046 sur l'échantillon de test.

Le modèle est donc capable de généraliser sans problèmes de sur- ou de sous-apprentissage notables.

La Figure 59 présente les performances obtenues pour chacune des classes, et met en lumière la difficulté du modèle à gérer la classe 3 (des personnes tuées lors de l'accident), qui est largement minoritaire dans le jeu de données. La précision de cette classe n'est pas bonne (0.13) puisque le modèle prédit comme tués de nombreux accidentés des classes 0, 1 et 2. Le rappel de la classe 3 est également relativement moyen (0.45), le modèle ayant tendance à positionner les personnes réellement tuées (classe 3) dans la classe des blessés hospitalisés (classe 2). La distinction entre ces 2 classes peut effectivement être relativement difficile et potentiellement relever de caractéristiques qui ne figurent pas dans la base de données mise à notre disposition (antécédents médicaux des accidentés par exemple).

	precision	recall	f1-score	support
0	0.71	0.81	0.76	46137
1	0.68	0.48	0.56	45097
2	0.41	0.40	0.41	17500
3	0.13	0.45	0.20	3050
accuracy			0.60	111784
macro avg	0.48	0.54	0.48	111784
weighted avg	0.64	0.60	0.61	111784

		Classes prédites			
Classes réelles	0	0	1	2	3
		0	37237	4836	2055
1	13037	21571	6953	3536	
2	1923	4932	7077	3568	
3	256	392	1019	1383	

(a) (b)

Figure 59 : Métriques (a) et matrice de confusion (b) du modèle de régression logistique de référence

III.1.2 Optimisation du modèle

A. Ajustement de paramètres

	Méthode de validation	Référence	Nouveau modèle	Décision
Ajout d'un undersampler	Validation croisée (cv=3)	0.5990 ± 0.0046	0.5659 ± 0.0054	Pas d'undersampler

Tableau 5 : Méthodes d'ajustement des paramètres

B. Ajustement des hyperparamètres penalty et C avec GridSearchCV

A été également étudiée l'influence de :

- la pénalité (norme L1 ou L2) : le solveur par défaut ne pouvant pas gérer la norme L1, le solveur 'liblinear' a été utilisé avec la pénalité 'L1', et 'newton-cg' avec 'L2',
- du type d'approche (multi_class) : « one-vs-rest » ou « auto »
- du paramètre de régularisation C : valeurs étudiées : [0.01, 0.1, 1, 10].

mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_log_reg_C	param_log_reg_multi_class	param_log_reg_penalty	param_log_reg_solver	param	std	test_score	split01	test_score	mean_test_score	std_test_score	rank	test_score	split00	train_score	mean_train_score	std_train_score
0.22342634	0.176944	0.377859	0.046371	0.01	auto	l1	liblinear	{'log_reg_C': 0.01, 'log_reg_multi_class': 'ovr'}	0.618109	0.617691	0.617900	0.000209	7	0.617459	0.619880	0.618869	0.000211			
1.2274962	0.374311	0.498839	0.126826	0.01	ovr	l1	liblinear	{'log_reg_C': 0.01, 'log_reg_multi_class': 'ovr'}	0.618109	0.617691	0.617900	0.000209	7	0.617459	0.619880	0.618869	0.000211			
2.99351320	22.533772	0.291428	0.0005734	0.1	auto	l1	liblinear	{'log_reg_C': 0.1, 'log_reg_multi_class': 'ovr'}	0.620715	0.619522	0.620119	0.000596	5	0.620411	0.621878	0.621144	0.000734			
3.9183268	18.478648	0.348932	0.0005681	0.1	ovr	l1	liblinear	{'log_reg_C': 0.1, 'log_reg_multi_class': 'ovr'}	0.620715	0.619522	0.620119	0.000596	5	0.620411	0.621878	0.621144	0.000734			
147.105681	35.949050	0.2478147	0.156132	1	auto	l1	liblinear	{'log_reg_C': 1, 'log_reg_multi_class': 'ovr'}	0.621067	0.619415	0.620481	0.000826	1	0.620351	0.621854	0.621103	0.000751			
148.266555	36.339669	0.2483576	0.1111029	1	ovr	l1	liblinear	{'log_reg_C': 1, 'log_reg_multi_class': 'ovr'}	0.621067	0.619415	0.620481	0.000826	1	0.620351	0.621814	0.621103	0.000751			
59.365097	48.943746	0.2814477	0.049160	10	auto	l1	liblinear	{'log_reg_C': 10, 'log_reg_multi_class': 'ovr'}	0.621043	0.619524	0.620148	0.000895	3	0.620411	0.621729	0.621070	0.000659			
7.57366793	47.878939	0.2374766	0.0256003	10	ovr	l1	liblinear	{'log_reg_C': 10, 'log_reg_multi_class': 'ovr'}	0.621043	0.619524	0.620148	0.000895	3	0.620411	0.621729	0.621070	0.000659			
8.18.65111	1.750830	0.377120	0.091809	1	auto	l2	newton-cg	{'log_reg_C': 1, 'log_reg_multi_class': 'ovr'}	0.576072	0.570905	0.570788	0.000116	10	0.571865	0.572557	0.572211	0.000346			
13.575491	0.588455	0.232525	0.064079	1	ovr	l2	newton-cg	{'log_reg_C': 1, 'log_reg_multi_class': 'ovr'}	0.599096	0.599651	0.599373	0.000277	9	0.599299	0.601320	0.600310	0.000111			

Figure 60 : Résultats du GridSearchCV sur penalty, le type d'approche et C, hyperparamètres de la régression logistique

La Figure 60 souligne que, de façon générale, la pénalité L1 (avec le solveur liblinear) conduit à de meilleurs résultats que la pénalité L2. Parmi les coefficients de régularisation analysés, C=1 est celui ayant conduit aux meilleurs résultats, avec un paramètre « multi-class » réglé sur 'auto'.

Les paramètres du modèle optimisé sont donc récapitulés dans le Tableau 6 et les résultats de ce modèle sur l'échantillon de test sont donnés dans la Figure 61, en termes de métriques et de matrice de confusion.

penalty	'l1'	intercept_scaling	1	multi_class	'auto'
dual	False	class_weight	'balanced'	verbose	0
tol	0.0001	random_state	None	warm_start	False
C	1	solver	'liblinear'	n_jobs	None
fit_intercept	True	max_iter	5000	l1_ratio	None

Tableau 6 : Paramètres du modèle de régression logistique après optimisation

	precision	recall	f1-score	support	
	Indemnes	0.70	0.82	0.76	46137
Blessés légers	0.66	0.54	0.59	45097	
Blessés graves	0.45	0.36	0.40	17500	
Tués	0.16	0.37	0.22	3050	
accuracy			0.62	111784	
macro avg	0.49	0.52	0.49	111784	
weighted avg	0.63	0.62	0.62	111784	

Classes réelles	Classes prédictes			
	0	1	2	3
0	37858	5392	1671	1216
1	13543	24283	4960	2311
2	2252	6354	6298	2596
3	333	557	1019	1141

(a)

(b)

Figure 61 : Métriques (a) et matrice de confusion (b) du modèle de régression logistique optimisé

L'amélioration globale n'est pas flagrante. On note une amélioration des précisions des classes 2 (blessés graves) et 3 (tués) (aux dépens des rappels de ces classes), et un f1-score légèrement amélioré pour la classe 3 (tués).

III.1.3 Interprétabilité des résultats

A. Analyse des coefficients de la régression logistique

La Figure 62 présente les boîtes à moustaches des coefficients de la régression logistique (modèle optimisé) obtenus lors d'une validation croisée à 3 échantillons. Il ressort de cette figure la forte influence de :

- Place_rec_4.0 : le fait d'être piéton
- Obsm_1.0 : obstacle mobile heurté de type piéton
- Les catégories de véhicules, dans l'ordre, vélos, motos, poids lourds, autres, transports en commun et voitures,
- Situ_5.0 : la présence sur une piste cyclable

On remarque que les valeurs absolues des coefficients associés aux variables continues restent relativement faibles. Il est envisageable que cela soit un inconvénient de la régression logistique, qui, cherchant des relations linéaires entre variables, ne permet pas de faire ressortir une influence potentiellement non linéaire des variables continues.

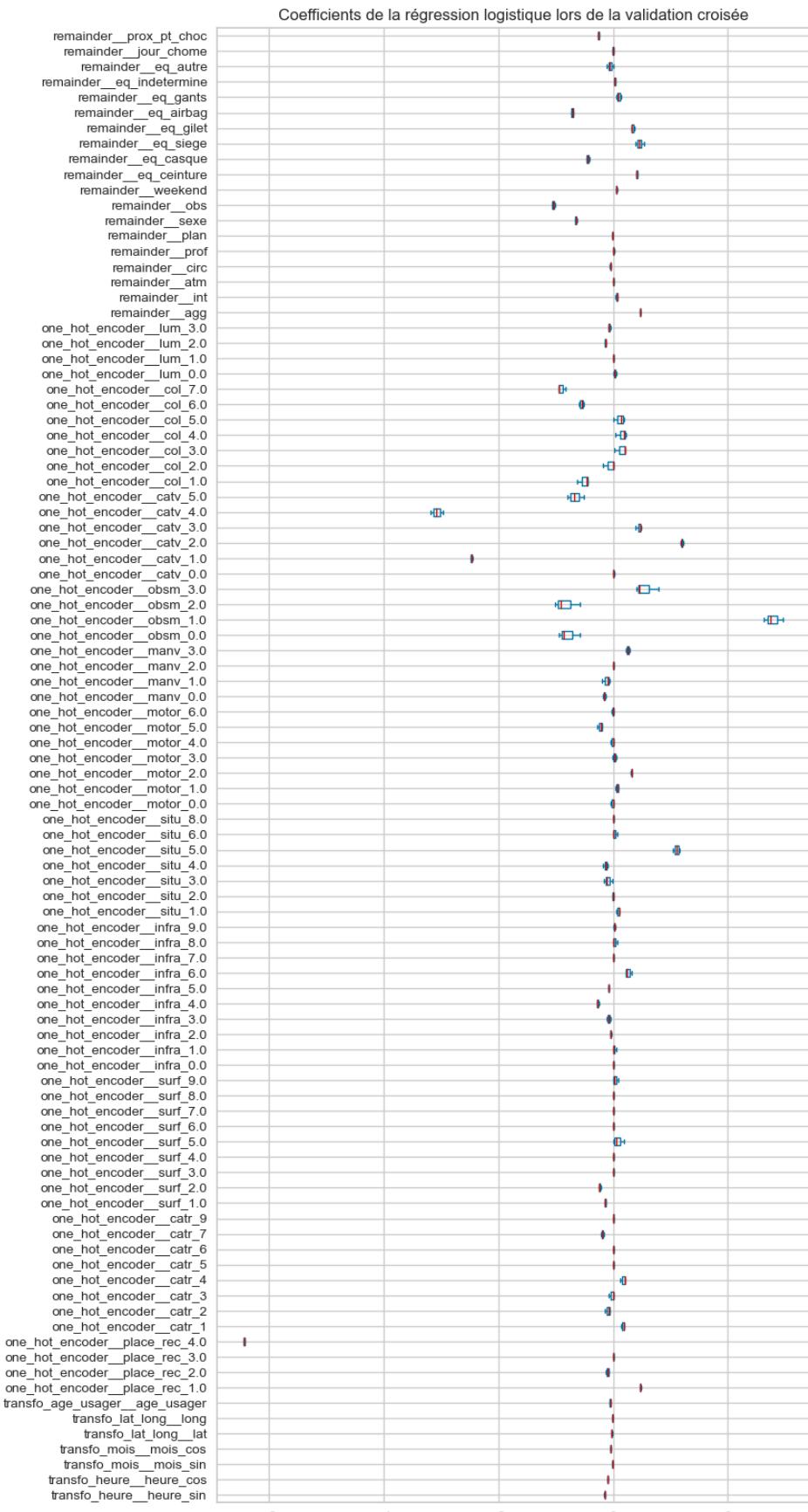


Figure 62 : Coefficients du modèle de régression logistique optimisé lors de la validation croisée

B. Analyse factorielle de données mixtes

Une analyse factorielle de données mixtes a été menée à l'aide de la bibliothèque prince.

La Figure 63 présente l'évolution du pourcentage de variance expliquée en fonction du nombre de composantes. La courbe associée à l'axe des ordonnées de gauche donne le cumul de variance expliquée en fonction du nombre de composantes retenues, tandis que l'histogramme, associé à l'axe des ordonnées de droite, renseigne sur la contribution de chaque composante.

Cette analyse confirme la complexité de notre jeu de données puisqu'il faut une trentaine de composantes pour atteindre 80% de la variance expliquée, et laisse supposer l'absence dans la base de variables explicatives fortement informatives sur la variance de notre jeu de données.

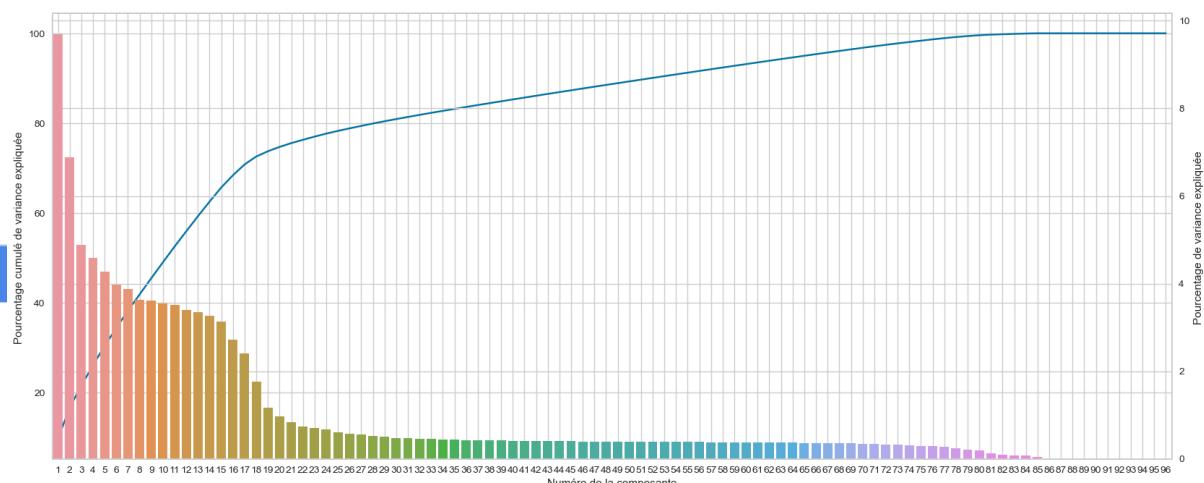


Figure 63 : Variance expliquée, en cumul (axe de gauche) ou unitairement (axe de droite), en fonction du nombre de composantes retenues

A titre d'informations, la Figure 64 présente les variables qui contribuent le plus fortement aux deux premières composantes de l'analyse factorielle (qui ne permettent cependant d'expliquer que 15% de la variance de notre jeu de données)

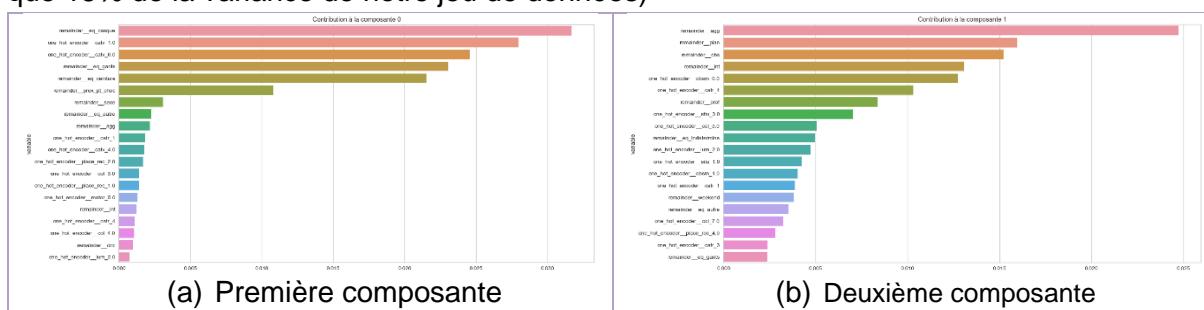


Figure 64 : Les 20 variables contribuant le plus aux deux premières composantes de l'analyse factorielle

III.2 Support Vector Machine

III.2.1 Modèle de référence

A. Description du modèle

La complexité des SVM est comprise entre $\mathcal{O}(n_{variables} \times n_{observations}^2)$ et $\mathcal{O}(n_{variables} \times n_{observations}^3)$, ce qui est difficilement envisageable pour notre problème présentant une centaine de variables et près de 500 000 observations. Il est alors plutôt recommandé de considérer un modèle de type LinearSVC ou SGDClassifier, après éventuellement une approximation de type Nyström pour introduire des non-linéarités dans le modèle.

Le modèle est construit sous forme de pipeline en 3 étapes :

- Etape 1 : préprocessor de type ColumnTransformer où sont appliquées les étapes de preprocessing décidées à l'issue de l'analyse préliminaire des données :
 - o Transformation des heures et des mois
 - o Transformation des latitudes et longitudes (RobustScaler)
 - o Transformation de l'âge (MinMaxScaler)
 - o Encodage des variables catégorielles (one_hot_encoder)
 - o Passage des autres variables de la base de données (remainder)
- Etape 2 : approximation de Nyström
- Etape 3 : modèle de type Linear SVC

B. Paramètres

Nyström	kernel	'rbf'	gamma	None	coef0	None
	degree	2	kernel_params	None	n_components	300
	random_state	None	n_jobs	None		
LinearSVC	penalty	'l2'	C	1.0	multi_class	'ovr'
	loss	'hinge'	fit_intercept	True	verbose	0
	dual	'warn'	class_weight	'balanced'	intercept_scaling	1
	tol	0.0001	random_state	42	max_iter	1000

Tableau 7 : Paramètres de l'approximation Nyström et du modèle de LinearSVC pris comme référence

Une validation croisée sur 3 échantillons conduit à un score de :

- 0.6021 ± 0.0029 sur l'échantillon d'apprentissage,
- et de 0.5979 ± 0.0022 sur l'échantillon de test.

Le modèle est capable de généraliser sans sur- ou de sous-apprentissage notable.

III.2.2 Optimisation du modèle - Ajustement des hyperparamètres penalty, loss, C et multi_class avec Optuna

Les influences suivantes ont été regardées, à savoir celles de :

- du degré de l'approximation : entre 2 et 4,
- du nombre de composantes de l'approximation : entre 150 et 450,
- la pénalité (norme L1 ou L2),
- du paramètre de régularisation C : valeurs étudiées : entre 0.05 et 10,
- du paramètre multi_class ('ovr' ou 'crammer_singer'),

A l'issue de l'optimisation, les meilleurs paramètres parmi ceux testés sont les suivants :

degree	2	C	1.2480028741569764
n_components	436	multi_class	Ovr
		penalty	L2

Les résultats de ce modèle sur l'échantillon de test sont donnés dans la Figure 65, en termes de métriques et de matrice de confusion. Ils sont relativement comparables à ceux obtenus par la régression logistique. **Le recours à l'approximation Nyström, telle qu'implémentée ici, n'a pas sensiblement permis d'améliorer les performances du modèle de classification.**

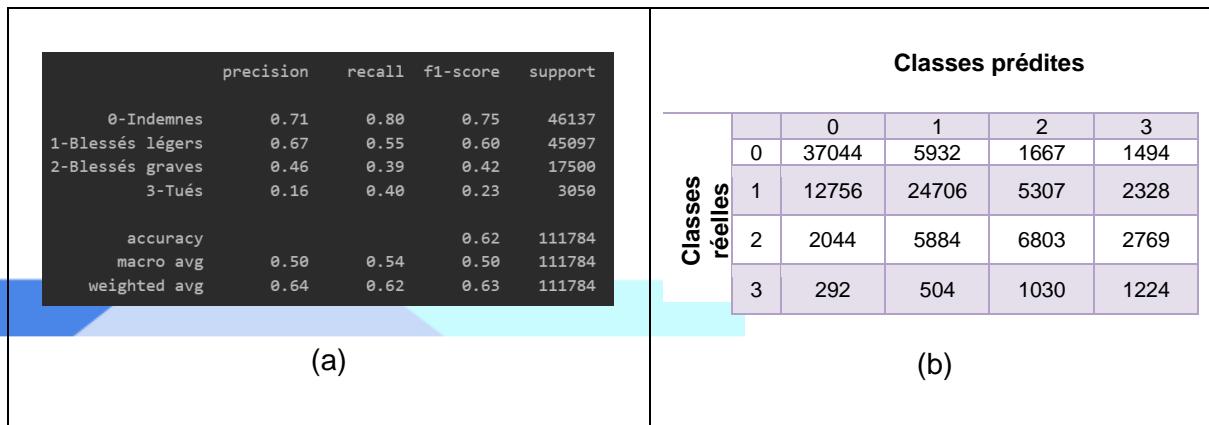


Figure 65 : Métriques (a) et matrice de confusion (b) du modèle LinearSVC précédé d'une approximation de Nyström

III.3 Decision Tree Classifier - Random Forest Classifier - Balanced Random Forest

Traités de la même manière, ces 3 modèles sont donc présentés ensemble.

III.3.1 Modèle de référence

A. Description du modèle

Pour modéliser, nous reprenons le jeu de données sans dummys que nous avons créé lors du nettoyage des données. Nous supprimons les variables 'an', 'jour', 'grav-rec', 'date' et 'dep' que nous avions conservées pour les visualisations et les séries temporelles.

Les arbres de décision et les random forest ne nécessitent pas de traitement particulier :

- les heures, mois, l'âge, les latitudes et longitudes sont conservées sans normalisation,
- les variables catégorielles sont conservées sans faire de dummys.

B. Paramètres

Pour chacun des algorithmes, nous réalisons une première modélisation avec les paramètres natifs de chaque algorithme afin de calculer l'importance de chacune des 35 variables. Puis, nous entraînons à nouveau le modèle natif en ajoutant une par une les variables (dans l'ordre d'importance décroissant) dans le but de ne sélectionner que les variables qui maximisent l'accuracy. Ceci nous permet d'éliminer les variables suivantes (Figure 66) selon l'algorithme :

Decision Tree	Random Forest	Balanced Random Forest																																																																																																																																																																																																																								
Conservation de 31 variables	Conservation de 34 variables	Conservation de 29 variables																																																																																																																																																																																																																								
<table border="1"> <thead> <tr> <th>feature</th> <th>importance</th> </tr> </thead> <tbody> <tr><td>lat</td><td>0.131</td></tr> <tr><td>long</td><td>0.123</td></tr> <tr><td>age_usager</td><td>0.113</td></tr> <tr><td>eq_ceinture</td><td>0.095</td></tr> <tr><td>heure</td><td>0.068</td></tr> <tr><td>mois</td><td>0.057</td></tr> <tr><td>col</td><td>0.039</td></tr> <tr><td>place_rec</td><td>0.038</td></tr> <tr><td>obs</td><td>0.032</td></tr> <tr><td>agg</td><td>0.023</td></tr> <tr><td>obsm</td><td>0.023</td></tr> <tr><td>catr</td><td>0.021</td></tr> <tr><td>catv</td><td>0.021</td></tr> <tr><td>manv</td><td>0.019</td></tr> <tr><td>lum</td><td>0.015</td></tr> <tr><td>infra</td><td>0.015</td></tr> <tr><td>sexe</td><td>0.015</td></tr> <tr><td>circ</td><td>0.014</td></tr> <tr><td>motor</td><td>0.013</td></tr> <tr><td>weekend</td><td>0.013</td></tr> <tr><td>situ</td><td>0.012</td></tr> <tr><td>eq_inetermine</td><td>0.011</td></tr> <tr><td>jour_chome</td><td>0.011</td></tr> <tr><td>prox_pt_choc</td><td>0.011</td></tr> <tr><td>prof</td><td>0.011</td></tr> <tr><td>int</td><td>0.010</td></tr> <tr><td>plan</td><td>0.010</td></tr> <tr><td>surf</td><td>0.010</td></tr> <tr><td>atm</td><td>0.009</td></tr> <tr><td>eq_casque</td><td>0.007</td></tr> <tr><td>eq_airbag</td><td>0.003</td></tr> <tr><td>eq_gants</td><td>0.002</td></tr> <tr><td>eq_autre</td><td>0.002</td></tr> <tr><td>eq_siege</td><td>0.001</td></tr> <tr><td>eq_gilet</td><td>0.001</td></tr> </tbody> </table>	feature	importance	lat	0.131	long	0.123	age_usager	0.113	eq_ceinture	0.095	heure	0.068	mois	0.057	col	0.039	place_rec	0.038	obs	0.032	agg	0.023	obsm	0.023	catr	0.021	catv	0.021	manv	0.019	lum	0.015	infra	0.015	sexe	0.015	circ	0.014	motor	0.013	weekend	0.013	situ	0.012	eq_inetermine	0.011	jour_chome	0.011	prox_pt_choc	0.011	prof	0.011	int	0.010	plan	0.010	surf	0.010	atm	0.009	eq_casque	0.007	eq_airbag	0.003	eq_gants	0.002	eq_autre	0.002	eq_siege	0.001	eq_gilet	0.001	<table border="1"> <thead> <tr> <th>feature</th> <th>importance</th> </tr> </thead> <tbody> <tr><td>lat</td><td>0.108</td></tr> <tr><td>age_usager</td><td>0.104</td></tr> <tr><td>long</td><td>0.104</td></tr> <tr><td>heure</td><td>0.072</td></tr> <tr><td>mois</td><td>0.060</td></tr> <tr><td>eq_ceinture</td><td>0.049</td></tr> <tr><td>col</td><td>0.045</td></tr> <tr><td>place_rec</td><td>0.044</td></tr> <tr><td>obsm</td><td>0.033</td></tr> <tr><td>catv</td><td>0.032</td></tr> <tr><td>catr</td><td>0.028</td></tr> <tr><td>eq_casque</td><td>0.022</td></tr> <tr><td>manv</td><td>0.021</td></tr> <tr><td>lum</td><td>0.020</td></tr> <tr><td>motor</td><td>0.019</td></tr> <tr><td>obs</td><td>0.019</td></tr> <tr><td>sexe</td><td>0.018</td></tr> <tr><td>agg</td><td>0.018</td></tr> <tr><td>infra</td><td>0.017</td></tr> <tr><td>weekend</td><td>0.016</td></tr> <tr><td>jour_chome</td><td>0.015</td></tr> <tr><td>prox_pt_choc</td><td>0.015</td></tr> <tr><td>prof</td><td>0.014</td></tr> <tr><td>situ</td><td>0.014</td></tr> <tr><td>circ</td><td>0.013</td></tr> <tr><td>eq_ineterminate</td><td>0.013</td></tr> <tr><td>surf</td><td>0.013</td></tr> <tr><td>int</td><td>0.013</td></tr> <tr><td>atm</td><td>0.012</td></tr> <tr><td>plan</td><td>0.012</td></tr> <tr><td>eq_gants</td><td>0.008</td></tr> <tr><td>eq_airbag</td><td>0.003</td></tr> <tr><td>eq_gilet</td><td>0.002</td></tr> <tr><td>eq_autre</td><td>0.002</td></tr> <tr><td>eq_siege</td><td>0.001</td></tr> </tbody> </table>	feature	importance	lat	0.108	age_usager	0.104	long	0.104	heure	0.072	mois	0.060	eq_ceinture	0.049	col	0.045	place_rec	0.044	obsm	0.033	catv	0.032	catr	0.028	eq_casque	0.022	manv	0.021	lum	0.020	motor	0.019	obs	0.019	sexe	0.018	agg	0.018	infra	0.017	weekend	0.016	jour_chome	0.015	prox_pt_choc	0.015	prof	0.014	situ	0.014	circ	0.013	eq_ineterminate	0.013	surf	0.013	int	0.013	atm	0.012	plan	0.012	eq_gants	0.008	eq_airbag	0.003	eq_gilet	0.002	eq_autre	0.002	eq_siege	0.001	<table border="1"> <thead> <tr> <th>feature</th> <th>importance</th> </tr> </thead> <tbody> <tr><td>lat</td><td>0.105</td></tr> <tr><td>age_usager</td><td>0.102</td></tr> <tr><td>long</td><td>0.099</td></tr> <tr><td>heure</td><td>0.074</td></tr> <tr><td>mois</td><td>0.062</td></tr> <tr><td>col</td><td>0.049</td></tr> <tr><td>eq_ceinture</td><td>0.039</td></tr> <tr><td>catr</td><td>0.034</td></tr> <tr><td>place_rec</td><td>0.032</td></tr> <tr><td>obsm</td><td>0.027</td></tr> <tr><td>agg</td><td>0.026</td></tr> <tr><td>catv</td><td>0.025</td></tr> <tr><td>lum</td><td>0.025</td></tr> <tr><td>manv</td><td>0.023</td></tr> <tr><td>obs</td><td>0.022</td></tr> <tr><td>situ</td><td>0.020</td></tr> <tr><td>sexe</td><td>0.018</td></tr> <tr><td>infra</td><td>0.018</td></tr> <tr><td>weekend</td><td>0.018</td></tr> <tr><td>motor</td><td>0.017</td></tr> <tr><td>circ</td><td>0.016</td></tr> <tr><td>jour_chome</td><td>0.016</td></tr> <tr><td>prox_pt_choc</td><td>0.016</td></tr> <tr><td>prof</td><td>0.016</td></tr> <tr><td>plan</td><td>0.015</td></tr> <tr><td>surf</td><td>0.015</td></tr> <tr><td>int</td><td>0.015</td></tr> <tr><td>atm</td><td>0.014</td></tr> <tr><td>eq_casque</td><td>0.013</td></tr> <tr><td>eq_ineterminate</td><td>0.013</td></tr> <tr><td>eq_gants</td><td>0.006</td></tr> <tr><td>eq_airbag</td><td>0.003</td></tr> <tr><td>eq_autre</td><td>0.003</td></tr> <tr><td>eq_siege</td><td>0.001</td></tr> <tr><td>eq_gilet</td><td>0.001</td></tr> </tbody> </table>	feature	importance	lat	0.105	age_usager	0.102	long	0.099	heure	0.074	mois	0.062	col	0.049	eq_ceinture	0.039	catr	0.034	place_rec	0.032	obsm	0.027	agg	0.026	catv	0.025	lum	0.025	manv	0.023	obs	0.022	situ	0.020	sexe	0.018	infra	0.018	weekend	0.018	motor	0.017	circ	0.016	jour_chome	0.016	prox_pt_choc	0.016	prof	0.016	plan	0.015	surf	0.015	int	0.015	atm	0.014	eq_casque	0.013	eq_ineterminate	0.013	eq_gants	0.006	eq_airbag	0.003	eq_autre	0.003	eq_siege	0.001	eq_gilet	0.001
feature	importance																																																																																																																																																																																																																									
lat	0.131																																																																																																																																																																																																																									
long	0.123																																																																																																																																																																																																																									
age_usager	0.113																																																																																																																																																																																																																									
eq_ceinture	0.095																																																																																																																																																																																																																									
heure	0.068																																																																																																																																																																																																																									
mois	0.057																																																																																																																																																																																																																									
col	0.039																																																																																																																																																																																																																									
place_rec	0.038																																																																																																																																																																																																																									
obs	0.032																																																																																																																																																																																																																									
agg	0.023																																																																																																																																																																																																																									
obsm	0.023																																																																																																																																																																																																																									
catr	0.021																																																																																																																																																																																																																									
catv	0.021																																																																																																																																																																																																																									
manv	0.019																																																																																																																																																																																																																									
lum	0.015																																																																																																																																																																																																																									
infra	0.015																																																																																																																																																																																																																									
sexe	0.015																																																																																																																																																																																																																									
circ	0.014																																																																																																																																																																																																																									
motor	0.013																																																																																																																																																																																																																									
weekend	0.013																																																																																																																																																																																																																									
situ	0.012																																																																																																																																																																																																																									
eq_inetermine	0.011																																																																																																																																																																																																																									
jour_chome	0.011																																																																																																																																																																																																																									
prox_pt_choc	0.011																																																																																																																																																																																																																									
prof	0.011																																																																																																																																																																																																																									
int	0.010																																																																																																																																																																																																																									
plan	0.010																																																																																																																																																																																																																									
surf	0.010																																																																																																																																																																																																																									
atm	0.009																																																																																																																																																																																																																									
eq_casque	0.007																																																																																																																																																																																																																									
eq_airbag	0.003																																																																																																																																																																																																																									
eq_gants	0.002																																																																																																																																																																																																																									
eq_autre	0.002																																																																																																																																																																																																																									
eq_siege	0.001																																																																																																																																																																																																																									
eq_gilet	0.001																																																																																																																																																																																																																									
feature	importance																																																																																																																																																																																																																									
lat	0.108																																																																																																																																																																																																																									
age_usager	0.104																																																																																																																																																																																																																									
long	0.104																																																																																																																																																																																																																									
heure	0.072																																																																																																																																																																																																																									
mois	0.060																																																																																																																																																																																																																									
eq_ceinture	0.049																																																																																																																																																																																																																									
col	0.045																																																																																																																																																																																																																									
place_rec	0.044																																																																																																																																																																																																																									
obsm	0.033																																																																																																																																																																																																																									
catv	0.032																																																																																																																																																																																																																									
catr	0.028																																																																																																																																																																																																																									
eq_casque	0.022																																																																																																																																																																																																																									
manv	0.021																																																																																																																																																																																																																									
lum	0.020																																																																																																																																																																																																																									
motor	0.019																																																																																																																																																																																																																									
obs	0.019																																																																																																																																																																																																																									
sexe	0.018																																																																																																																																																																																																																									
agg	0.018																																																																																																																																																																																																																									
infra	0.017																																																																																																																																																																																																																									
weekend	0.016																																																																																																																																																																																																																									
jour_chome	0.015																																																																																																																																																																																																																									
prox_pt_choc	0.015																																																																																																																																																																																																																									
prof	0.014																																																																																																																																																																																																																									
situ	0.014																																																																																																																																																																																																																									
circ	0.013																																																																																																																																																																																																																									
eq_ineterminate	0.013																																																																																																																																																																																																																									
surf	0.013																																																																																																																																																																																																																									
int	0.013																																																																																																																																																																																																																									
atm	0.012																																																																																																																																																																																																																									
plan	0.012																																																																																																																																																																																																																									
eq_gants	0.008																																																																																																																																																																																																																									
eq_airbag	0.003																																																																																																																																																																																																																									
eq_gilet	0.002																																																																																																																																																																																																																									
eq_autre	0.002																																																																																																																																																																																																																									
eq_siege	0.001																																																																																																																																																																																																																									
feature	importance																																																																																																																																																																																																																									
lat	0.105																																																																																																																																																																																																																									
age_usager	0.102																																																																																																																																																																																																																									
long	0.099																																																																																																																																																																																																																									
heure	0.074																																																																																																																																																																																																																									
mois	0.062																																																																																																																																																																																																																									
col	0.049																																																																																																																																																																																																																									
eq_ceinture	0.039																																																																																																																																																																																																																									
catr	0.034																																																																																																																																																																																																																									
place_rec	0.032																																																																																																																																																																																																																									
obsm	0.027																																																																																																																																																																																																																									
agg	0.026																																																																																																																																																																																																																									
catv	0.025																																																																																																																																																																																																																									
lum	0.025																																																																																																																																																																																																																									
manv	0.023																																																																																																																																																																																																																									
obs	0.022																																																																																																																																																																																																																									
situ	0.020																																																																																																																																																																																																																									
sexe	0.018																																																																																																																																																																																																																									
infra	0.018																																																																																																																																																																																																																									
weekend	0.018																																																																																																																																																																																																																									
motor	0.017																																																																																																																																																																																																																									
circ	0.016																																																																																																																																																																																																																									
jour_chome	0.016																																																																																																																																																																																																																									
prox_pt_choc	0.016																																																																																																																																																																																																																									
prof	0.016																																																																																																																																																																																																																									
plan	0.015																																																																																																																																																																																																																									
surf	0.015																																																																																																																																																																																																																									
int	0.015																																																																																																																																																																																																																									
atm	0.014																																																																																																																																																																																																																									
eq_casque	0.013																																																																																																																																																																																																																									
eq_ineterminate	0.013																																																																																																																																																																																																																									
eq_gants	0.006																																																																																																																																																																																																																									
eq_airbag	0.003																																																																																																																																																																																																																									
eq_autre	0.003																																																																																																																																																																																																																									
eq_siege	0.001																																																																																																																																																																																																																									
eq_gilet	0.001																																																																																																																																																																																																																									

Figure 66 : Variables sélectionnées selon le modèle avec leur importance

En utilisant ce nombre de variables réduit, nous obtenons les rapports de classification et les matrices de confusions suivant (Figure 67) :

Decision Tree						
(Train accuracy = 99.9% - Test accuracy = 57.02%)						
	precision	recall	f1-score	support		
1	0.68	0.68	0.68	46137	1	31143
2	0.16	0.17	0.16	3050	2	427
3	0.38	0.38	0.38	17500	3	2740
4	0.56	0.57	0.56	45097	4	11536
accuracy			0.57	111784		6857
macro avg	0.44	0.45	0.45	111784		25480
weighted avg	0.57	0.57	0.57	111784		

Random Forest						
(Train accuracy = 99.9% - Test accuracy = 66.96%)						
	precision	recall	f1-score	support		
1	0.72	0.82	0.77	46137	1	37922
2	0.43	0.05	0.10	3050	2	397
3	0.52	0.41	0.46	17500	3	2405
4	0.65	0.66	0.65	45097	4	11633
accuracy			0.67	111784		7855
macro avg	0.58	0.48	0.49	111784		29659
weighted avg	0.66	0.67	0.66	111784		

Balanced Random Forest						
(Train accuracy = 66.69% - Test accuracy = 59.04%)						
	precision	recall	f1-score	support		
1	0.73	0.80	0.76	46137	1	36943
2	0.13	0.66	0.22	3050	2	182
3	0.39	0.42	0.41	17500	3	1649
4	0.72	0.44	0.54	45097	4	12149
accuracy			0.59	111784		2711
macro avg	0.49	0.58	0.48	111784		19682
weighted avg	0.66	0.59	0.60	111784		

Figure 67 : Métriques et matrice de confusion selon le modèle

Dans tous les cas, on remarque la difficulté des modèles à gérer la classe 2 (celles des personnes tuées) :

- Decision Tree a une précision et un recall équilibré mais très faible,
- Random Forest a un f1-score plus faible (précision beaucoup plus élevée mais au détriment du recall),
- Balanced Random Forest a un f1-score augmenté (recall augmenté au détriment de la précision).

De plus, Decision Tree et Random Forest présentent un fort sur apprentissage.

III.3.2 Optimisation du modèle

Dans un premier temps, pour chacun des algorithmes, nous allons rechercher avec un grid search les paramètres qui maximisent l'accuracy. Pour cela chaque modèle est défini de la manière suivante (Tableau 8) :

Decision Tree		Random Forest		Balanced random Forest	
criterion	['gini', 'entropy']	criterion	['gini', 'entropy']	criterion	['gini', 'entropy']
max_depth	[10, 20, None]	max_depth	[10, 20, None]	max_depth	[10, 20, None]
max_features	['sqrt']	min_samples_leaf	[1, 10, 20]	min_samples_leaf	[1, 10, 20]
min_samples_leaf	[1, 10, 20]	min_samples_split	[2, 10, 20]	min_samples_split	[2, 10, 20]
min_samples_split	[2, 10, 20]	random_state	[42]	random_state	[42]
random_state	[42]	n_estimators	[100]	n_estimators	[100]
splitter	['best', 'random']	bootstrap	[False, True]	bootstrap	[True]
		n_jobs	[-1]	n_jobs	[-1]
				sampling_strategy	['all']
				remplacement	[True]

Tableau 8 : Choix d'optimisation des paramètres selon le modèle

Puis nous avons entraîné chacun de ces modèles avec :

- soit ajout d'un paramètre supplémentaire : weight_class défini à None, 'balanced' ou un dictionnaire de poids que nous avons optimisé,
- soit sur un jeu de données où l'on fait de l'oversampling avec SMOTENC ou RandomOverSampler (non réalisé pour Balanced Random Forest),
- soit sur un jeu de données où l'on fait de l'undersampling avec RandomUnderSampler ou ClusterCentroids (non réalisé pour Balanced Random Forest).

Enfin, dans chaque cas, nous recherchons le meilleur max_depth en réalisant un grid search avec un scoring basé sur le f1-score-macro.

III.3.3 Résultats

Pour chacun de ces algorithmes, nous choisissons le meilleur résultat en nous basant sur les modèles qui permettent d'optimiser le macro avg. Voici les rapports de classification et les matrices de confusions (Figure 68) :

Decision Tree																																																																												
(Train accuracy = 65.54% - Test accuracy = 62.78%)																																																																												
Paramètres :																																																																												
{'class_weight': {1: 1, 2: 4, 3: 1, 4: 1}, 'criterion': 'gini', 'max_depth': 12, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'random_state': 42, 'splitter': 'best'}																																																																												
<table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>1</td><td>0.71</td><td>0.81</td><td>0.75</td><td>46137</td></tr> <tr> <td>2</td><td>0.16</td><td>0.38</td><td>0.23</td><td>3050</td></tr> <tr> <td>3</td><td>0.49</td><td>0.25</td><td>0.33</td><td>17500</td></tr> <tr> <td>4</td><td>0.64</td><td>0.61</td><td>0.62</td><td>45097</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.63</td><td>111784</td></tr> <tr> <td>macro avg</td><td>0.50</td><td>0.51</td><td>0.48</td><td>111784</td></tr> <tr> <td>weighted avg</td><td>0.63</td><td>0.63</td><td>0.62</td><td>111784</td></tr> </tbody> </table>					precision	recall	f1-score	support	1	0.71	0.81	0.75	46137	2	0.16	0.38	0.23	3050	3	0.49	0.25	0.33	17500	4	0.64	0.61	0.62	45097	accuracy			0.63	111784	macro avg	0.50	0.51	0.48	111784	weighted avg	0.63	0.63	0.62	111784	<table border="1"> <thead> <tr> <th>Classes prédictes</th><th>1</th><th>2</th><th>3</th><th>4</th></tr> </thead> <tbody> <tr> <td>Clases réelles</td><td></td><td></td><td></td><td></td></tr> <tr> <td>1</td><td>37306</td><td>816</td><td>853</td><td>7162</td></tr> <tr> <td>2</td><td>358</td><td>1151</td><td>758</td><td>783</td></tr> <tr> <td>3</td><td>2372</td><td>3103</td><td>4419</td><td>7606</td></tr> <tr> <td>4</td><td>12733</td><td>2003</td><td>3055</td><td>27306</td></tr> </tbody> </table>			Classes prédictes	1	2	3	4	Clases réelles					1	37306	816	853	7162	2	358	1151	758	783	3	2372	3103	4419	7606	4	12733	2003	3055	27306
	precision	recall	f1-score	support																																																																								
1	0.71	0.81	0.75	46137																																																																								
2	0.16	0.38	0.23	3050																																																																								
3	0.49	0.25	0.33	17500																																																																								
4	0.64	0.61	0.62	45097																																																																								
accuracy			0.63	111784																																																																								
macro avg	0.50	0.51	0.48	111784																																																																								
weighted avg	0.63	0.63	0.62	111784																																																																								
Classes prédictes	1	2	3	4																																																																								
Clases réelles																																																																												
1	37306	816	853	7162																																																																								
2	358	1151	758	783																																																																								
3	2372	3103	4419	7606																																																																								
4	12733	2003	3055	27306																																																																								
Random Forest																																																																												
(Train accuracy = 81.09% - Test accuracy = 65.22%)																																																																												
Paramètres :																																																																												
{'bootstrap': True, 'class_weight': 'balanced', 'criterion': 'entropy', 'max_depth': 18, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100, 'n_jobs': -1, 'random_state': 42}																																																																												
<table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>1</td><td>0.71</td><td>0.85</td><td>0.77</td><td>46137</td></tr> <tr> <td>2</td><td>0.24</td><td>0.25</td><td>0.24</td><td>3050</td></tr> <tr> <td>3</td><td>0.45</td><td>0.53</td><td>0.49</td><td>17500</td></tr> <tr> <td>4</td><td>0.71</td><td>0.53</td><td>0.61</td><td>45097</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.65</td><td>111784</td></tr> <tr> <td>macro avg</td><td>0.53</td><td>0.54</td><td>0.53</td><td>111784</td></tr> <tr> <td>weighted avg</td><td>0.66</td><td>0.65</td><td>0.65</td><td>111784</td></tr> </tbody> </table>					precision	recall	f1-score	support	1	0.71	0.85	0.77	46137	2	0.24	0.25	0.24	3050	3	0.45	0.53	0.49	17500	4	0.71	0.53	0.61	45097	accuracy			0.65	111784	macro avg	0.53	0.54	0.53	111784	weighted avg	0.66	0.65	0.65	111784	<table border="1"> <thead> <tr> <th>Classes prédictes</th><th>1</th><th>2</th><th>3</th><th>4</th></tr> </thead> <tbody> <tr> <td>Clases réelles</td><td></td><td></td><td></td><td></td></tr> <tr> <td>1</td><td>39000</td><td>361</td><td>2191</td><td>4585</td></tr> <tr> <td>2</td><td>311</td><td>771</td><td>1470</td><td>498</td></tr> <tr> <td>3</td><td>2163</td><td>1434</td><td>9225</td><td>4678</td></tr> <tr> <td>4</td><td>13103</td><td>683</td><td>7407</td><td>23904</td></tr> </tbody> </table>			Classes prédictes	1	2	3	4	Clases réelles					1	39000	361	2191	4585	2	311	771	1470	498	3	2163	1434	9225	4678	4	13103	683	7407	23904
	precision	recall	f1-score	support																																																																								
1	0.71	0.85	0.77	46137																																																																								
2	0.24	0.25	0.24	3050																																																																								
3	0.45	0.53	0.49	17500																																																																								
4	0.71	0.53	0.61	45097																																																																								
accuracy			0.65	111784																																																																								
macro avg	0.53	0.54	0.53	111784																																																																								
weighted avg	0.66	0.65	0.65	111784																																																																								
Classes prédictes	1	2	3	4																																																																								
Clases réelles																																																																												
1	39000	361	2191	4585																																																																								
2	311	771	1470	498																																																																								
3	2163	1434	9225	4678																																																																								
4	13103	683	7407	23904																																																																								
Balanced Random Forest																																																																												
(Train accuracy = 68.38% - Test accuracy = 60.71%)																																																																												
Paramètres :																																																																												
{'bootstrap': True, 'class_weight': None, 'criterion': 'entropy', 'max_depth': 30, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100, 'n_jobs': -1, 'random_state': 42, 'replacement': True, 'sampling_strategy': 'all'}																																																																												
<table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>1</td><td>0.72</td><td>0.81</td><td>0.76</td><td>46137</td></tr> <tr> <td>2</td><td>0.16</td><td>0.54</td><td>0.24</td><td>3050</td></tr> <tr> <td>3</td><td>0.40</td><td>0.49</td><td>0.44</td><td>17500</td></tr> <tr> <td>4</td><td>0.72</td><td>0.45</td><td>0.55</td><td>45097</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.61</td><td>111784</td></tr> <tr> <td>macro avg</td><td>0.50</td><td>0.57</td><td>0.50</td><td>111784</td></tr> <tr> <td>weighted avg</td><td>0.66</td><td>0.61</td><td>0.61</td><td>111784</td></tr> </tbody> </table>					precision	recall	f1-score	support	1	0.72	0.81	0.76	46137	2	0.16	0.54	0.24	3050	3	0.40	0.49	0.44	17500	4	0.72	0.45	0.55	45097	accuracy			0.61	111784	macro avg	0.50	0.57	0.50	111784	weighted avg	0.66	0.61	0.61	111784	<table border="1"> <thead> <tr> <th>Classes prédictes</th><th>1</th><th>2</th><th>3</th><th>4</th></tr> </thead> <tbody> <tr> <td>Clases réelles</td><td></td><td></td><td></td><td></td></tr> <tr> <td>1</td><td>37557</td><td>1491</td><td>2639</td><td>4450</td></tr> <tr> <td>2</td><td>194</td><td>1655</td><td>966</td><td>235</td></tr> <tr> <td>3</td><td>1757</td><td>4213</td><td>8492</td><td>3038</td></tr> <tr> <td>4</td><td>12614</td><td>3134</td><td>9189</td><td>20160</td></tr> </tbody> </table>			Classes prédictes	1	2	3	4	Clases réelles					1	37557	1491	2639	4450	2	194	1655	966	235	3	1757	4213	8492	3038	4	12614	3134	9189	20160
	precision	recall	f1-score	support																																																																								
1	0.72	0.81	0.76	46137																																																																								
2	0.16	0.54	0.24	3050																																																																								
3	0.40	0.49	0.44	17500																																																																								
4	0.72	0.45	0.55	45097																																																																								
accuracy			0.61	111784																																																																								
macro avg	0.50	0.57	0.50	111784																																																																								
weighted avg	0.66	0.61	0.61	111784																																																																								
Classes prédictes	1	2	3	4																																																																								
Clases réelles																																																																												
1	37557	1491	2639	4450																																																																								
2	194	1655	966	235																																																																								
3	1757	4213	8492	3038																																																																								
4	12614	3134	9189	20160																																																																								

Figure 68 : Métriques et matrice de confusion avec les paramètres optimisés selon le modèle

Finalement, nous choisissons l'algorithme qui donne les meilleurs résultats parmi ces 3 modèles : le modèle utilisant l'algorithme Random Forest (Figure 69). Cependant, il y a un fort sur-apprentissage. Nous décidons donc de diminuer le max_depth ce qui nous permet d'obtenir le résultat suivant :

Random Forest																																																																																	
(Train accuracy = 65.52% - Test accuracy = 61.22%)																																																																																	
Paramètres :																																																																																	
{'bootstrap': True, 'class_weight': 'balanced', 'criterion': 'entropy', 'max_depth': 13, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100, 'n_jobs': -1, 'random_state': 42}																																																																																	
<table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>1</td><td>0.70</td><td>0.85</td><td>0.77</td><td>46137</td></tr> <tr> <td>2</td><td>0.17</td><td>0.49</td><td>0.26</td><td>3050</td></tr> <tr> <td>3</td><td>0.41</td><td>0.49</td><td>0.45</td><td>17500</td></tr> <tr> <td>4</td><td>0.74</td><td>0.42</td><td>0.54</td><td>45097</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.61</td><td>111784</td></tr> <tr> <td>macro avg</td><td>0.51</td><td>0.56</td><td>0.50</td><td>111784</td></tr> <tr> <td>weighted avg</td><td>0.66</td><td>0.61</td><td>0.61</td><td>111784</td></tr> </tbody> </table>				precision	recall	f1-score	support	1	0.70	0.85	0.77	46137	2	0.17	0.49	0.26	3050	3	0.41	0.49	0.45	17500	4	0.74	0.42	0.54	45097	accuracy			0.61	111784	macro avg	0.51	0.56	0.50	111784	weighted avg	0.66	0.61	0.61	111784	<table border="1"> <thead> <tr> <th></th><th>Classes prédites</th><th>1</th><th>2</th><th>3</th><th>4</th></tr> <tr> <th>Clases réelles</th><th></th><th></th><th></th><th></th><th></th></tr> </thead> <tbody> <tr> <td>1</td><td>39232</td><td>1262</td><td>2339</td><td>3304</td><td></td></tr> <tr> <td>2</td><td>288</td><td>1488</td><td>1016</td><td>258</td><td></td></tr> <tr> <td>3</td><td>2225</td><td>3578</td><td>8632</td><td>3065</td><td></td></tr> <tr> <td>4</td><td>14677</td><td>2275</td><td>9058</td><td>19087</td><td></td></tr> </tbody> </table>				Classes prédites	1	2	3	4	Clases réelles						1	39232	1262	2339	3304		2	288	1488	1016	258		3	2225	3578	8632	3065		4	14677	2275	9058	19087	
	precision	recall	f1-score	support																																																																													
1	0.70	0.85	0.77	46137																																																																													
2	0.17	0.49	0.26	3050																																																																													
3	0.41	0.49	0.45	17500																																																																													
4	0.74	0.42	0.54	45097																																																																													
accuracy			0.61	111784																																																																													
macro avg	0.51	0.56	0.50	111784																																																																													
weighted avg	0.66	0.61	0.61	111784																																																																													
	Classes prédites	1	2	3	4																																																																												
Clases réelles																																																																																	
1	39232	1262	2339	3304																																																																													
2	288	1488	1016	258																																																																													
3	2225	3578	8632	3065																																																																													
4	14677	2275	9058	19087																																																																													

Figure 69 : Métriques et matrice de confusion pour le meilleur modèle

En diminuant le sur apprentissage, on peut observer que la prédiction pour les indemnes et les tués (classe 1 et 2) est améliorée au détriment des blessés hospitalisés et légers (classes 3 et 4).

III.3.4 Interprétabilité des résultats

A. Analyse de l'importance donnée par le meilleur modèle

Le meilleur modèle utilisant l'algorithme Random forest est entraîné avec 34 variables dont voici le graphique selon l'importance donnée par le modèle (Figure 70) :

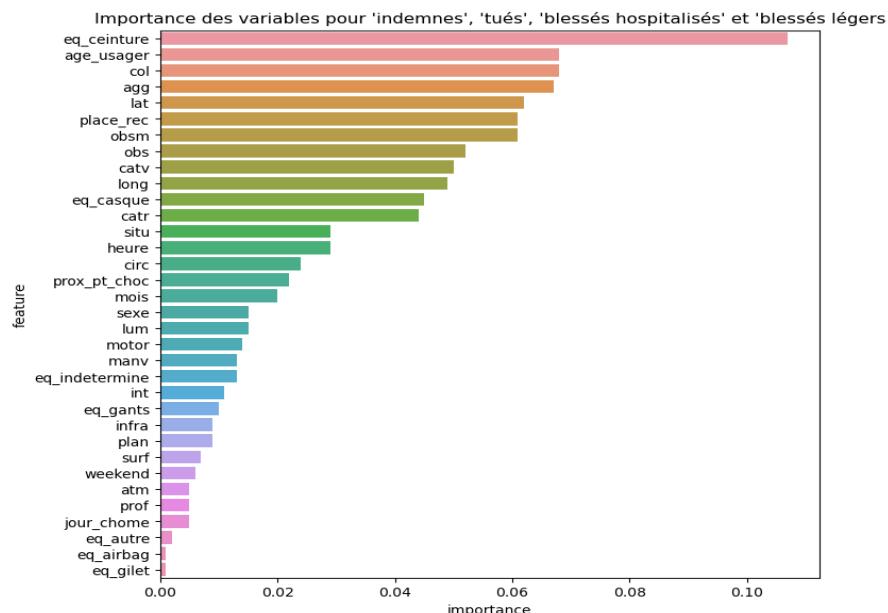


Figure 70 : Importance des variables pour le meilleur modèle Random Forest

Le modèle accorde une grande importance aux variables suivantes :

- très forte importance de l'utilisation ou non de la **ceinture de sécurité**,
- forte importance de l'**âge de l'usager**, du **type de collision** et du fait de **rouler en agglomération ou non**,
- importance légèrement moindre pour la **latitude**, la **place dans le véhicule ou piéton** et **l'obstacle mobile heurté**.

B. Analyse de l'interprétation du meilleur modèle avec SHAP

L'analyse de l'interprétation avec SHAP (Shapley additive explanations) permet d'attribuer à chaque variable une valeur d'importance pour une prédiction particulière et donc d'affiner la compréhension de la part de chaque variable dans la prédiction. Ceci nous permet d'obtenir les graphiques d'importance des variables (Figure 71), mais également les graphiques de densité des valeurs SHAP pour chaque variable (Figure 72) en fonction de la modalité de la variable cible 'grav'.

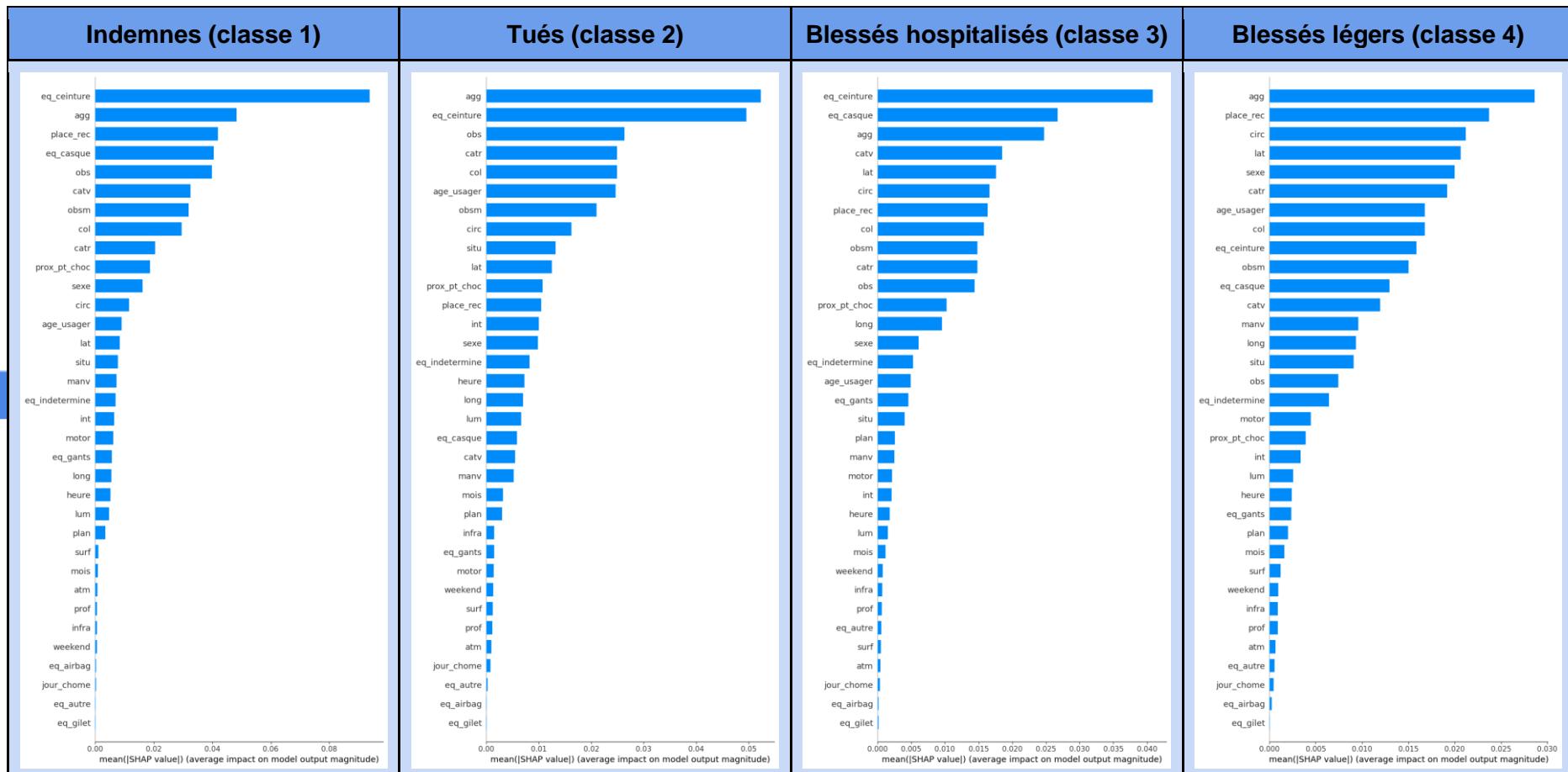


Figure 71 : Graphiques d'importance des variables selon SHAP

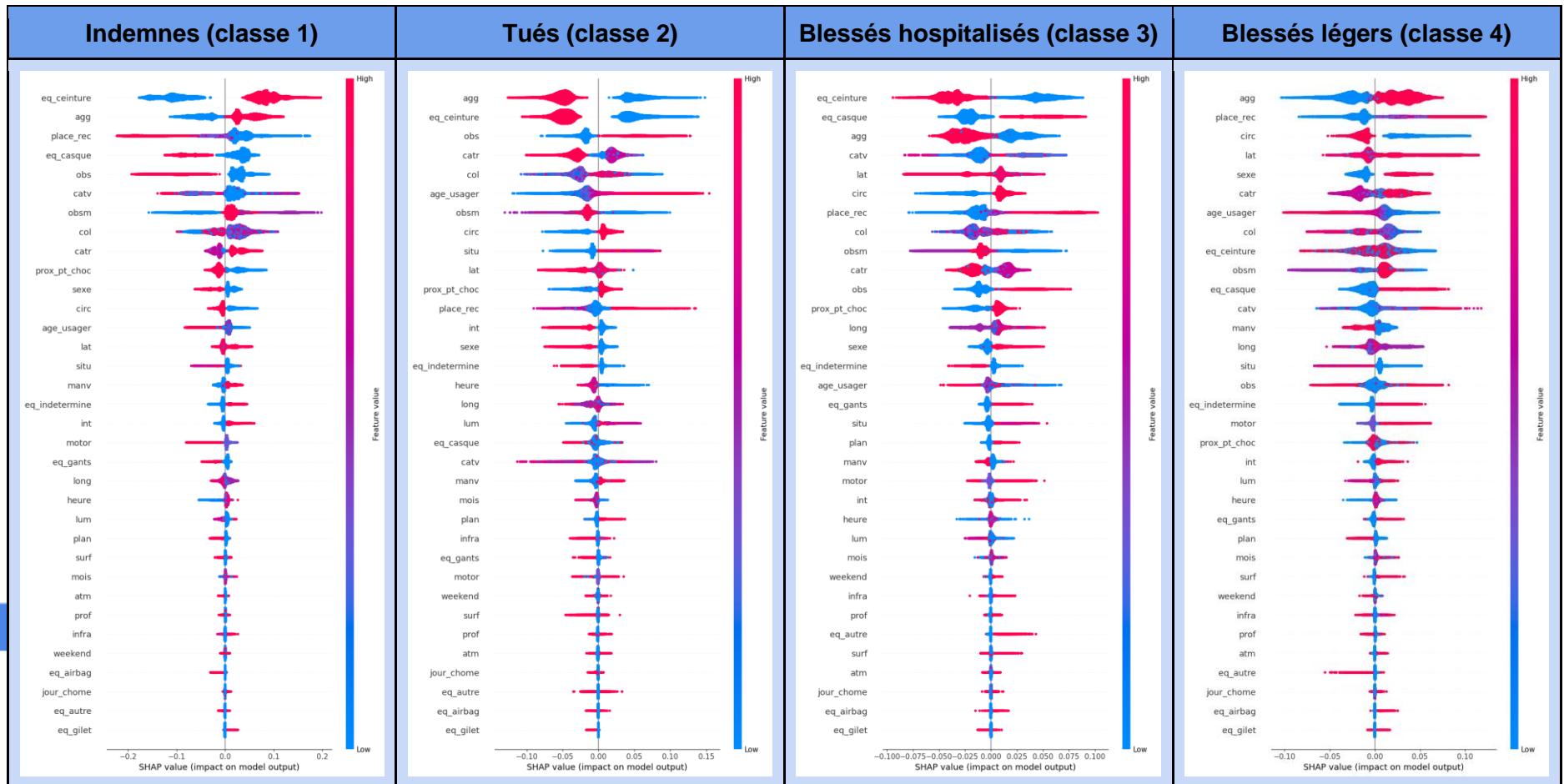


Figure 72 : Graphiques de densité des valeurs de SHAP

L'analyse de ces graphiques, nous permet de prévoir les variables qui influent sur le fait d'être classés indemne, tué, blessé hospitalisé ou blessé léger. Ainsi, en analysant les 5 premières variables les plus importantes selon SHAP, nous pouvons dire que les paramètres qui influent positivement sur la classification sont :

- **pour les indemnes :**
 - l'utilisation de la ceinture de sécurité,
 - rouler en agglomération,
 - être à l'avant du véhicule (les valeurs les plus faibles de la variable catégorielle 'place_rec' sont soit conducteur et/ou passager avant),
 - ne pas porter de casque (certainement pour le fait de ne pas être en vélo/trottinette ou moto),
 - ne pas heurter d'obstacle fixe.
- **pour les tués :**
 - rouler hors d'agglomération,
 - ne pas utiliser de ceinture de sécurité,
 - heurter un obstacle fixe.

Pour les variables 'catr' (variable catégorielle regroupant 8 modalités) et 'col' (variable catégorielle regroupant 7 modalités) la distinction entre les valeurs sur le graphique de densité des valeurs est plus floue et ne permet pas d'établir formellement quelles modalités influent positivement sur la classification.

- **pour les blessés hospitalisés :**
 - ne pas utiliser de ceinture de sécurité,
 - porter un casque (certainement pour le fait d'être en vélo/trottinette ou moto),
 - rouler hors agglomération.

Pour les variables 'catv' (variable catégorielle regroupant 6 modalités) et 'lat' (variable numérique) la distinction entre les valeurs sur le graphique de densité des valeurs est plus floue et ne permet pas d'établir formellement quelles modalités ou valeurs influent positivement sur la classification.

- **pour les blessés légers :**
 - rouler en agglomération,
 - être à l'arrière du véhicule ou piéton (les valeurs les plus élevées de la variable catégorielle 'place_rec' sont soit passager arrière, soit piéton),
 - rouler sur une route unidirectionnelle,
 - être une femme.

Pour la variable 'lat' (variable numérique) la distinction entre les valeurs sur le graphique de densité des valeurs est plus floue et ne permet pas d'établir formellement quelles valeurs influent positivement sur la classification.

III.4 CatBoost Classifier

III.4.1 Modèle de référence

A. Description du modèle

CatBoost est l'une des librairies permettant d'entraîner des arbres de décision pour la classification. Il prend en charge des données catégorielles (Cat) et utilise le Gradient Boosting (Boost).

B. Paramètres

Ce modèle s'appuie sur une centaine de paramètres pouvant être regroupés en différentes familles (le lecteur trouvera la liste complète sur la documentation à l'adresse suivante : https://catboost.ai/en/docs/concepts/python-reference_catboostclassifier) :

- Ceux classiques pour un problème de machine learning : le taux d'apprentissage (learning_rate), la fonction de perte (loss_function), le nombre d'itérations (iterations)...
- Ceux contrôlant l'échantillonnage des données pour chaque arbre : choix aléatoire (bagging_temperature)
- Ceux contrôlant la structure de l'arbre : sa symétrie (grow_policy), sa profondeur (depth) ...
- Ceux contrôlant la sélection des variables pour la construction des arbres (colsample_bylevel)
- Ceux régissant la pénalisation du modèle : (l2_leaf_reg)
- Ceux permettant de détecter le surapprentissage.

III.4.2 Optimisation du modèle

L'optimisation des hyperparamètres a été effectuée avec la librairie Optuna. L'espace d'hyperparamètres à tester a été le suivant :

- un nombre d'itérations compris entre 250 et 400,
- une fonction de perte parmi : « MultiClass » et « MultiClassOneVsAll »,
- un taux d'apprentissage compris entre 0.01 et 0.5,
- une profondeur de l'arbre comprise entre 4 et 10,
- un paramètre de pénalisation entre 5 et 10,
- une structure de l'arbre choisie parmi : « SymmetricTree » et « Lossguide »,
- une fraction de variables à choisir dans le processus de division, prise entre 0.05 et 1.

La métrique de performance choisie est le F1-score 'MICRO', avec pour ambition sa maximisation. L'optimisation bayésienne a été retenue pour la sélection des hyperparamètres dans l'espace.

La Figure 73 présente l'influence des hyperparamètres sur la performance du modèle pour une sélection de 20 jeux de données différents. Ce graphique met en lumière qu'il est plutôt bénéfique de choisir **des profondeurs importantes** et **des facteurs de pénalisation supérieurs à 7**. De façon générale, pour les méthodes impliquant des arbres, il apparaît que certes, l'augmentation des profondeurs améliore les performances sur l'échantillon test, mais cela accentue également le phénomène de sur-apprentissage.

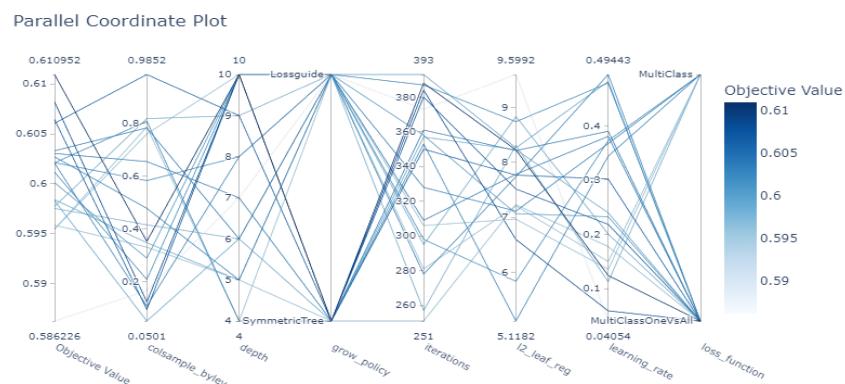


Figure 73 : Graphique à coordonnées parallèles présentant le lien entre les hyperparamètres de Catboost et la performance du modèle.

Les performances du meilleur modèle parmi ceux testés dans le processus d'optimisation sont données à la Figure 74. Encore une fois, **le potentiel gain de performance par optimisation du modèle reste limité**. Comparativement au meilleur modèle retenu jusqu'à présent, à savoir le Random Forest dont les résultats sont présentés à la Figure 69, le modèle Catboost permet une **amélioration des rappels des classes 3, celle des tués et 1, celle des blessés légers, mais au détriment de la précision sur ces classes et in fine, des f1-scores**. En conséquence, si l'on se fixe comme objectif de vouloir maximiser l'identification des tués parmi les accidentés de la route, alors le modèle CatBoost est un bon candidat. Néanmoins, à l'échelle de l'ensemble des classes, le modèle Random Forest apporte le meilleur compromis en termes de performance.

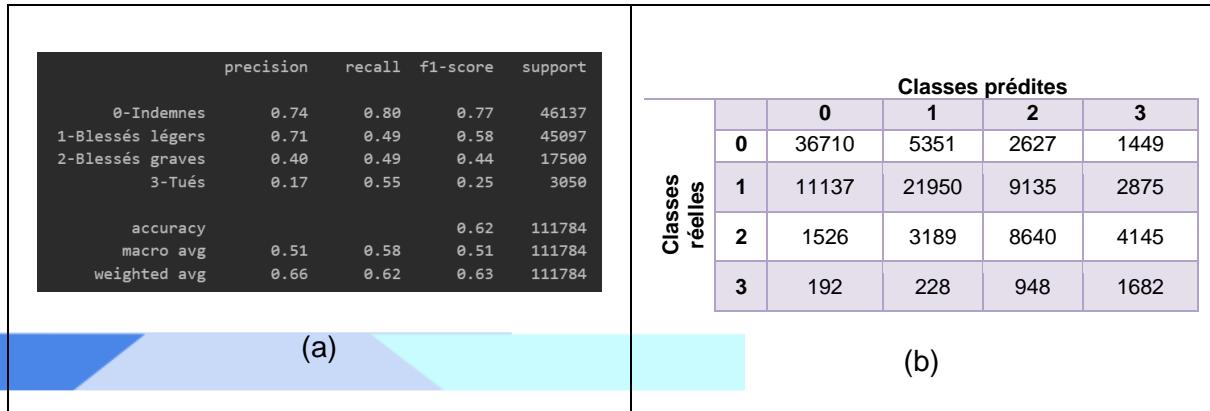


Figure 74 : Métriques (a) et matrice de confusion (b) du modèle CatBoost

III.4.3 Interprétabilité des résultats

A. Analyse de l'importance donnée par le meilleur modèle

La Figure 75 présente les variables les plus importantes pour le modèle CatBoost. Si l'ordre des variables n'est pas tout à fait identique à celui de la Figure 70 (où figure l'importance des variables dans le modèle Random Forest), les variables les plus influentes sont globalement similaires dans les deux approches : à savoir le port ou non de la ceinture de sécurité, le type de collision, la catégorie de véhicules, la latitude, la place occupée par l'usager (ou piéton), l'obstacle mobile heurté. Figure 75 : **Importance des variables pour le meilleur modèle CatBoost**

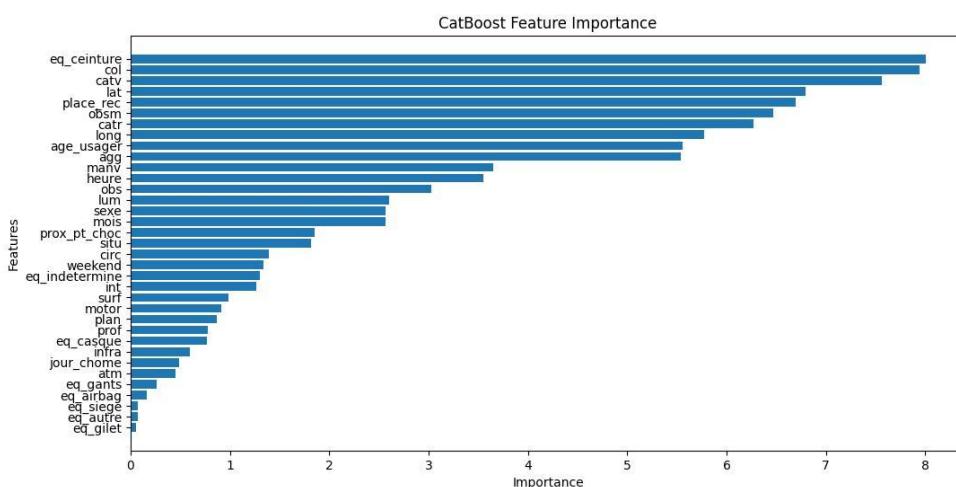


Figure 75 : Importance des variables pour le meilleur modèle CatBoost

B. Analyse de l'interprétation du meilleur modèle CatBoost avec SHAP

Les graphiques de densité des valeurs SHAP issus de l'analyse du modèle CatBoost (Figure 76) sont relativement comparables à ceux de la Figure 72 pour le modèle Random Forest.

Pour les indemnes, à l'exception du port du casque qui n'apparaît pas ici dans les variables les plus influentes, on retrouve la catégorie du véhicule impliqué, l'utilisation ou non de la ceinture de sécurité, le fait d'être à l'avant du véhicule, de ne pas heurter d'obstacle mobile, d'être dans un accident impliquant peu de véhicules.

Pour les tués, comme précédemment, le fait de ne pas porter de ceinture et d'être hors agglomération agrave les blessures. Les personnes les plus âgées sont plus susceptibles d'être tuées.

Pour les blessés hospitalisés, le non port de la ceinture, le fait d'être hors agglomération, d'occuper une place à l'avant sont des facteurs conduisant à l'appartenance à cette classe.

Enfin, les paramètres influents sur l'appartenance à la classe des blessés légers sont le fait d'être une femme, un passager arrière ou un piéton, d'heurter un obstacle mobile et d'être jeune.

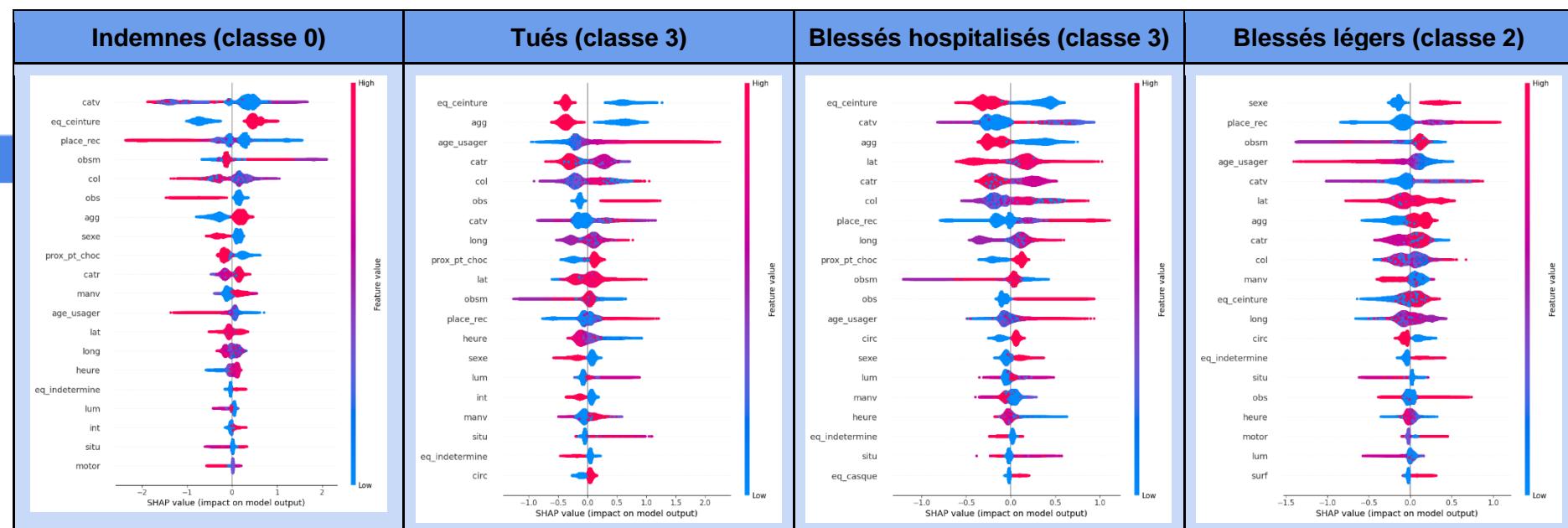


Figure 76 : Graphiques de densité des valeurs de SHAP pour le modèle optimisé CatBoost

III.5 XGBoost Classifier

III.5.1 Modèle de référence

A. Description du modèle

Pour cette partie de modélisation de référence qui permettra d'avoir d'une part une vision de la performance du modèle XGBoost sans hyper-paramétrage (du moins ceux par défaut) et d'autre part une base de comparaison avec les différentes modélisations optimisées, nous avons travaillé avec le jeu de données “data_cleaned_final_sans_dummies.csv” créé lors du preprocessing.

Nous avons supprimé les variables ‘jour’, ‘grav-rec’, ‘date’ et ‘prox_pt_choc’, qui représentent des variables sans signification ou redondantes.

Même si le modèle XGBoost ne souffre pas de l'obligation d'avoir des données normalisées, par cohérence nous avons appliqué une normalisation sur les variables ‘age-usager’, ‘latitude’, ‘longitude’ et ‘heure’.

Le dataset présentant un fort déséquilibre des classes pour la variable cible ‘grav’, nous avons décidé de tester un modèle **xgb_clf** selon 3 méthodes pour déterminer une base de référence :

1. Méthode 1 : Entraînement sur le dataset,
2. Méthode 2 : Application d'une *class_weight* lors du .fit() sur le dataset,
3. Méthode 3 : Application d'un rééchantillonnage sur le dataset.

B. Paramètres par défaut du modèle

n_estimators	100	subsample	1	reg_lambda	1
learning_rate	0.3	sampling_method	uniform	reg_alpha	0
gamma	0	colsample_bytree	1	tree_method	auto
max_depth	6	colsample_bylevel	1	scale_pos_weight	1
min_child_weight	1	colsample_bynode	1	refresh_leaf	1
process_type	default	max_leaves	0	num_parallel_tree	1
grow_policy	depthwise	max_bins	256	multi_strategy	one_output_per_tree
objective	softmax				

Tableau 9 : Paramètres par défaut de XGBoost

Quelques informations sur la répartition des valeurs de y_test :

```
y_test.value_counts()  
✓ 0.0s  
  
grav  
0    36910  
1    36078  
2    14000  
3    2440  
Name: count, dtype: int64
```

C. Courbes d'apprentissage

Application d'un early_stopping_rounds = 10 et eval_metric = ['merror', 'mlogloss'] et d'un seed = 42 avec évaluation eval_set=[(X_train, y_train), (X_test, y_test)].

Visualisation des courbes mlogloss et mrror suivant le nombre d'arbres (n_estimators) (Figure 77) :

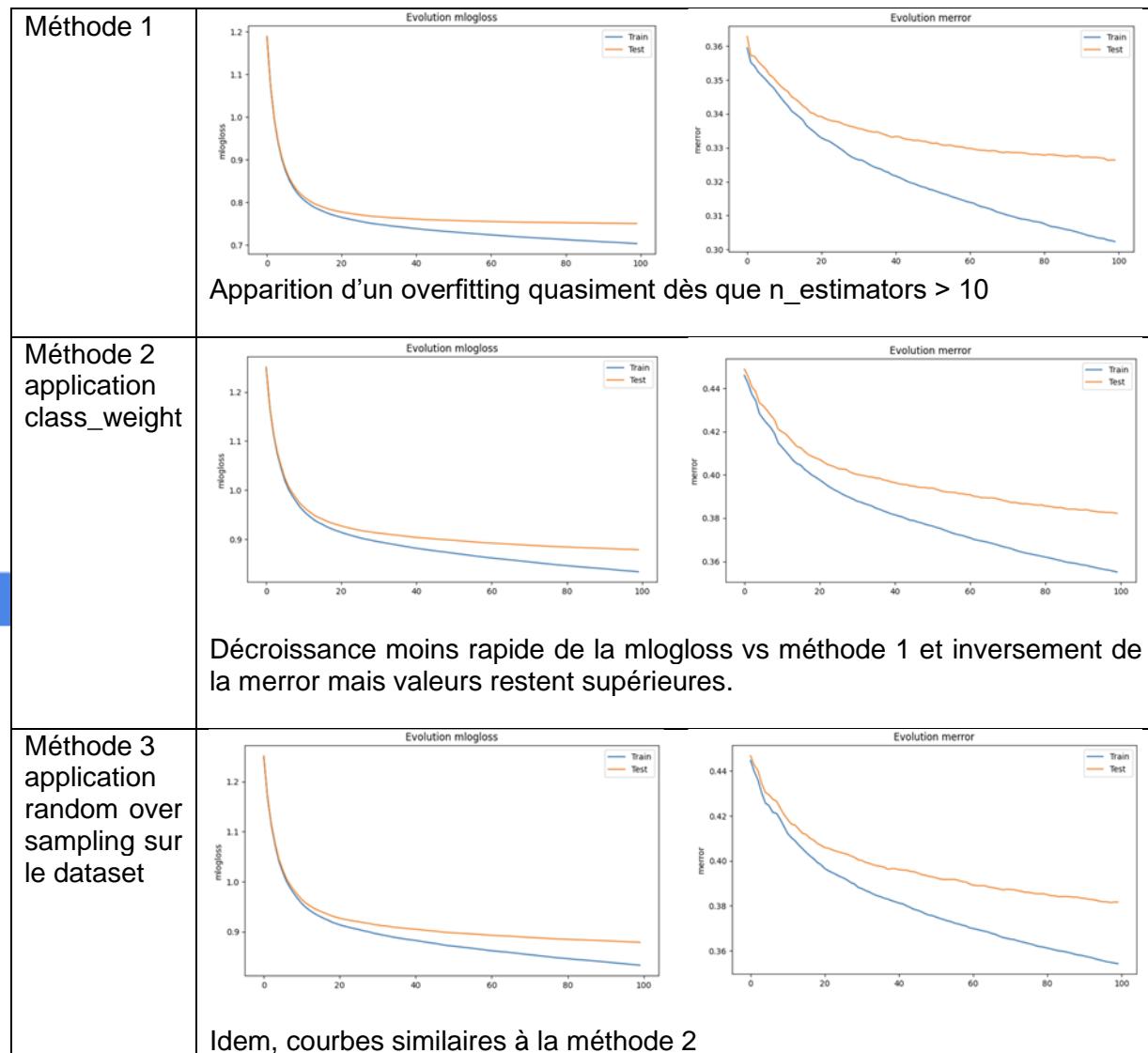


Figure 77 : Courbes de mlogloss et mrror suivant le nombre d'arbres

D. Evaluations

- Matrice de confusion

Méthode 1				Méthode 2				Méthode 3						
prédiction	0	1	2	3	prédiction	0	1	2	3	prédiction	0	1	2	3
réalité					réalité					réalité				
0	30135	5828	927	20	0	28789	4633	2185	1303	0	28809	4697	2105	1299
1	8991	23704	3325	58	1	8322	18292	6792	2672	1	8258	18443	6714	2663
2	1746	5884	6215	155	2	1175	2510	6780	3535	2	1164	2597	6662	3577
3	291	612	1351	186	3	151	173	728	1388	3	156	181	718	1385

Classe 3 très peu détectée et
confusion avec les autres classes (Faux Négatifs). Beaucoup de Faux Positifs mais plutôt fiable dans la prédiction de la classe 3. Résultat quasi-similaire à la méthode 2.

Figure 78 : Matrices de confusion pour chaque méthode

- Key Metrics

Méthode 1	Méthode 2	Méthode 3
Accuracy: 0.67 Balanced Accuracy: 0.50	Accuracy: 0.62 Balanced Accuracy: 0.59	Accuracy: 0.62 Balanced Accuracy: 0.58
Micro Precision: 0.67 Micro Recall: 0.67 Micro F1-score: 0.67	Micro Precision: 0.62 Micro Recall: 0.62 Micro F1-score: 0.62	Micro Precision: 0.62 Micro Recall: 0.62 Micro F1-score: 0.62
Macro Precision: 0.59 Macro Recall: 0.50 Macro F1-score: 0.51	Macro Precision: 0.51 Macro Recall: 0.59 Macro F1-score: 0.51	Macro Precision: 0.51 Macro Recall: 0.58 Macro F1-score: 0.51
Weighted Precision: 0.66 Weighted Recall: 0.67 Weighted F1-score: 0.66	Weighted Precision: 0.67 Weighted Recall: 0.62 Weighted F1-score: 0.63	Weighted Precision: 0.67 Weighted Recall: 0.62 Weighted F1-score: 0.63

Tableau 10 : Métriques principales pour chaque méthode

La méthode 2 permet d'avoir la **meilleure “balanced accuracy”** et aussi le **meilleur Macro Recall**, c'est à dire la **meilleure proportion de cas correctement détectés** mais ceci au détriment d'une précision plus faible et donc d'un taux de FAUX POSITIF plus élevé.

Dans notre cas, il conviendra de **définir le coût entre les FAUX NÉGATIFS et les FAUX POSITIFS**. C'est à dire faut-il mieux prédire la gravité quitte à accentuer cette dernière ou limiter les prédictions graves quitte à en oublier ?

- Rapport de classification

Méthode 1	pre	rec	spe	f1	geo	iba	sup	
	Indemne	0.73	0.82	0.79	0.77	0.80	0.65	36910
Blessé Léger	0.66	0.66	0.77	0.66	0.71	0.50	36078	
Blessé Grave	0.53	0.44	0.93	0.48	0.64	0.39	14000	
Tué	0.44	0.08	1.00	0.13	0.28	0.07	2440	
avg ↓ total	0.66	0.67	0.81	0.66	0.73	0.53	89428	
Le f1_score de la classe 3 (Tués) est médiocre, impact du recall très faible								
Méthode 2	pre	rec	spe	f1	geo	iba	sup	
	Indemne	0.75	0.78	0.82	0.76	0.80	0.63	36910
Blessé Léger	0.71	0.51	0.86	0.59	0.66	0.42	36078	
Blessé Grave	0.41	0.48	0.87	0.44	0.65	0.41	14000	
Tué	0.16	0.57	0.91	0.24	0.72	0.50	2440	
avg ↓ total	0.67	0.62	0.85	0.63	0.72	0.51	89428	
Légère amélioration du f1_score de la classe 3 mais au détriment des autres classes.								
Méthode 3	pre	rec	spe	f1	geo	iba	sup	
	Indemne	0.75	0.78	0.82	0.77	0.80	0.64	36910
Blessé Léger	0.71	0.51	0.86	0.59	0.66	0.42	36078	
Blessé Grave	0.41	0.48	0.87	0.44	0.64	0.40	14000	
Tué	0.16	0.57	0.91	0.24	0.72	0.50	2440	
avg ↓ total	0.67	0.62	0.85	0.63	0.72	0.51	89428	
Les résultats sont identiques à la méthode 2.								

Figure 79 : Rapports de classification selon la méthode

Malgré une accuracy de 67% sur l'ensemble de test, la méthode 1 semble souffrir du fort déséquilibre de la variable cible du dataset. Les méthodes 2 et 3 avec une accuracy de 61% gèrent mieux ce déséquilibre.

Pour rappel : classe 0 = 41.3%, classe 1 = 40.3%, classe 2 = 15.7%, classe 3 = 2.7%.

- Tracés des features importance (20 premières) selon weight, gain et cover

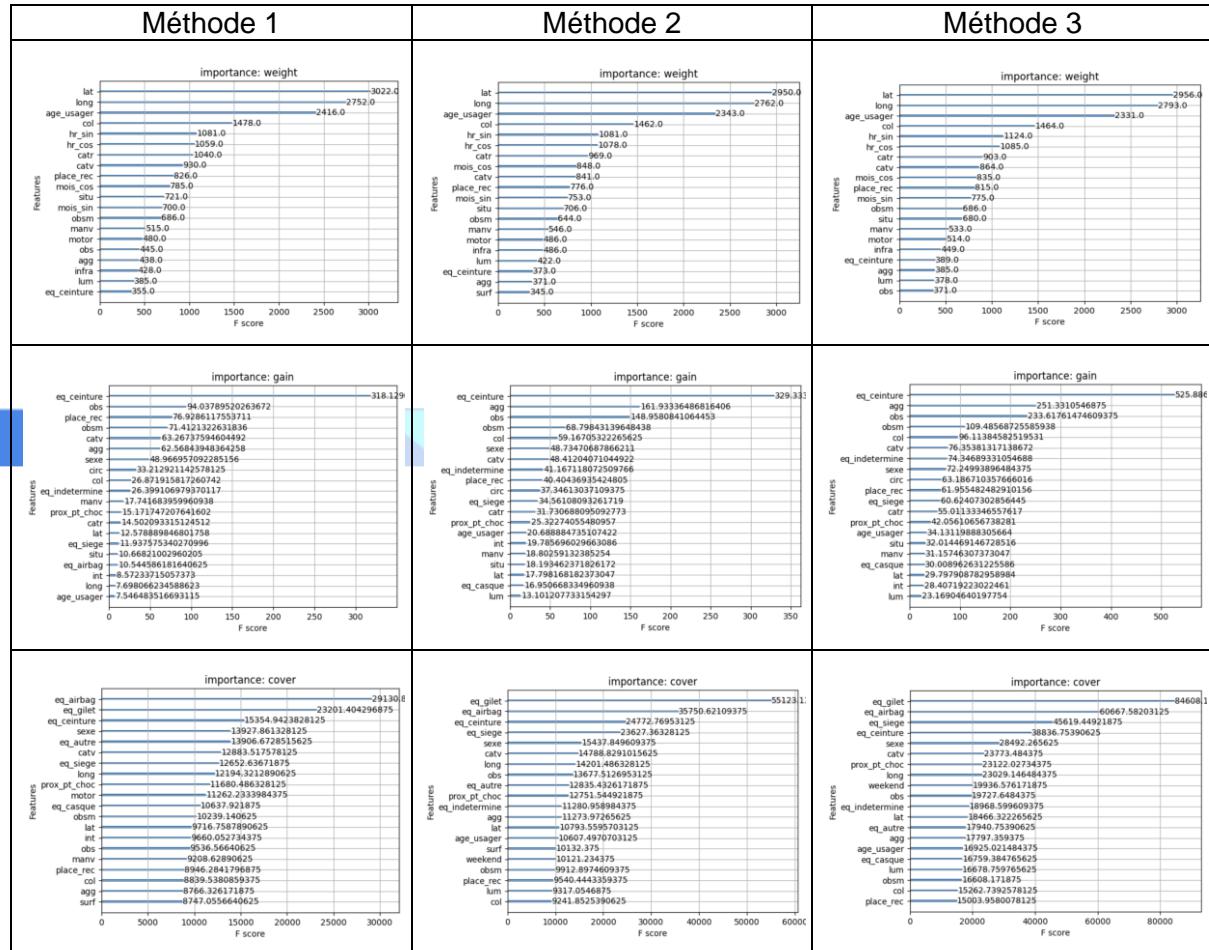


Figure 80 : Les 20 premières features importance selon weight, gain, cover et selon la méthode

Selon le *weight*, nous avons pour les 3 modèles les 6 features identiques dans le même ordre qui apparaissent dans la construction des arbres. Concernant le *gain* (réduction de la fonction de perte lors de l'utilisation d'une feature pour séparer une branche), on remarque l'importance de la feature 'eq_ceinture' qui est significativement plus élevée sur les 3 méthodes.

Pour terminer concernant le *cover* (utilisation de la feature pour la séparation des données), on remarque une rupture entre les 3 modèles.

En conclusion de cette première partie concernant l'évaluation du modèle de référence, nous retenons pour la suite, c'est-à-dire l'optimisation des hyperparamètres pour améliorer la performance, **la méthode 2 (ajout d'une class_weight)** qui semble être la plus pertinente par rapport à l'optimisation des métriques recherchées (precision et recall).

III.5.2 Optimisation de la méthode 2

Le modèle XGBoost étant un modèle comprenant beaucoup d'hyperparamètres, nous avons déployé une méthode agile utilisée par la communauté de Data Scientist pour économiser les temps de calcul, à savoir :

1. Fixer le learning_rate à 1,
2. Fixer le n_estimators puis optimiser les autres hyperparamètres,
3. Ajuster le learning_rate.

A. Paramètres optimisés

n_estimator	200	subsample	1	reg_lambda	1
learning_rate	0.05	sampling_method	uniform	reg_alpha	0
gamma	0	colsample_bytree	1	tree_method	auto
max_depth	3	colsample_bylevel	1	scale_pos_weight	1
min_child_weight	1	colsample_bynode	1	refresh_leaf	1
process_type	default	max_leaves	0	num_parallel_tree	1
grow_policy	depthwise	max_bins	256	multi_strategy	one_output_per_tree
objective	softmax				

Tableau 11 : Paramètres optimisés pour XGBoost

B. Courbe d'apprentissage

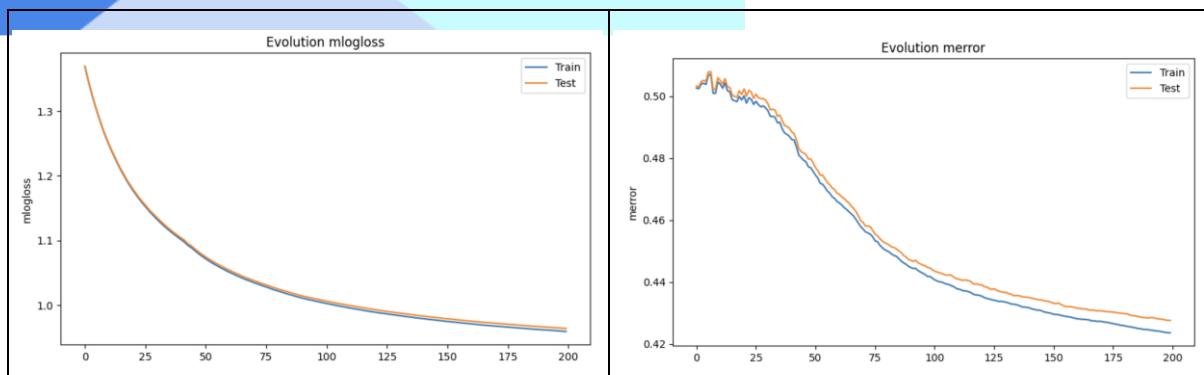


Figure 81 : Courbes de mlogloss et mrror pour le modèle optimisé

On peut constater la convergence entre les courbes de train et test sur les 2 métriques mlogloss et mrror.

C. Evaluation et comparaison

- Matrice de confusion

référence		model2f (modèle optimisé)			
prédiction	réalité	0	1	2	3
0	28789	4633	2185	1303	0
1	8322	18292	6792	2672	1
2	1175	2510	6780	3535	2
3	151	173	728	1388	3

On constate que la **classe 3 est bien plus détectée** au détriment d'un taux de faux positif plus élevé

Figure 82 : Matrices de confusion du modèle de référence et du modèle optimisé

- Rapport de classification

référence		pre	rec	spe	f1	geo	iba	sup
	Indemne	0.75	0.78	0.82	0.76	0.80	0.63	36910
	Blessé Léger	0.71	0.51	0.86	0.59	0.66	0.42	36078
	Blessé Grave	0.41	0.48	0.87	0.44	0.65	0.41	14000
	Tué	0.16	0.57	0.91	0.24	0.72	0.50	2440
	avg <i>L</i> total	0.67	0.62	0.85	0.63	0.72	0.51	89428
model2f		pre	rec	spe	f1	geo	iba	sup
	Indemne	0.71	0.79	0.77	0.75	0.78	0.62	36910
	Blessé Léger	0.70	0.41	0.88	0.51	0.60	0.34	36078
	Blessé Grave	0.38	0.41	0.88	0.39	0.60	0.34	14000
	Tué	0.12	0.63	0.88	0.21	0.74	0.54	2440
	avg <i>L</i> total	0.64	0.57	0.84	0.58	0.68	0.46	89428
	L'optimisation du recall de la classe Tué impacte fortement le recall des classes Blessé Léger et Blessé Grave et une perte de la précision sur les 4 classes donc un taux de FAUX POSITIFS important.							

Figure 83 : Rapports de classification pour le modèle de référence et le modèle optimisé

III.5.3 Interprétabilité des résultats avec SHAP

Les résultats de l'optimisation n'étant pas concluants dans le sens où l'optimisation n'a pas permis de faire un gain significatif, nous avons mené une étude d'interprétabilité pour comprendre comment le modèle prédit ses choix, quelles sont les features déterminantes. L'utilisation de SHAP nous permet de comprendre le fonctionnement du modèle en étudiant les graphiques d'importance des variables (Figure 84) et les graphiques de densité des valeurs SHAP pour chaque variable (Figure 85) en fonction de la variable cible.

La feature *eq_ceinture* est prépondérante sur les 4 classes de gravité notamment pour les indemnes donc l'utilisation de la ceinture joue un rôle important.

On remarque aussi que pour les classes Blessé Grave et Tué, les 3 features les plus importantes sont les mêmes, ce qui signifie une difficulté à séparer ces 2 classes avec certitude.

La feature *agg* (rouler en agglomération ou pas), induisant de façon sous-jacente une notion de vitesse, influe nettement sur la gravité.

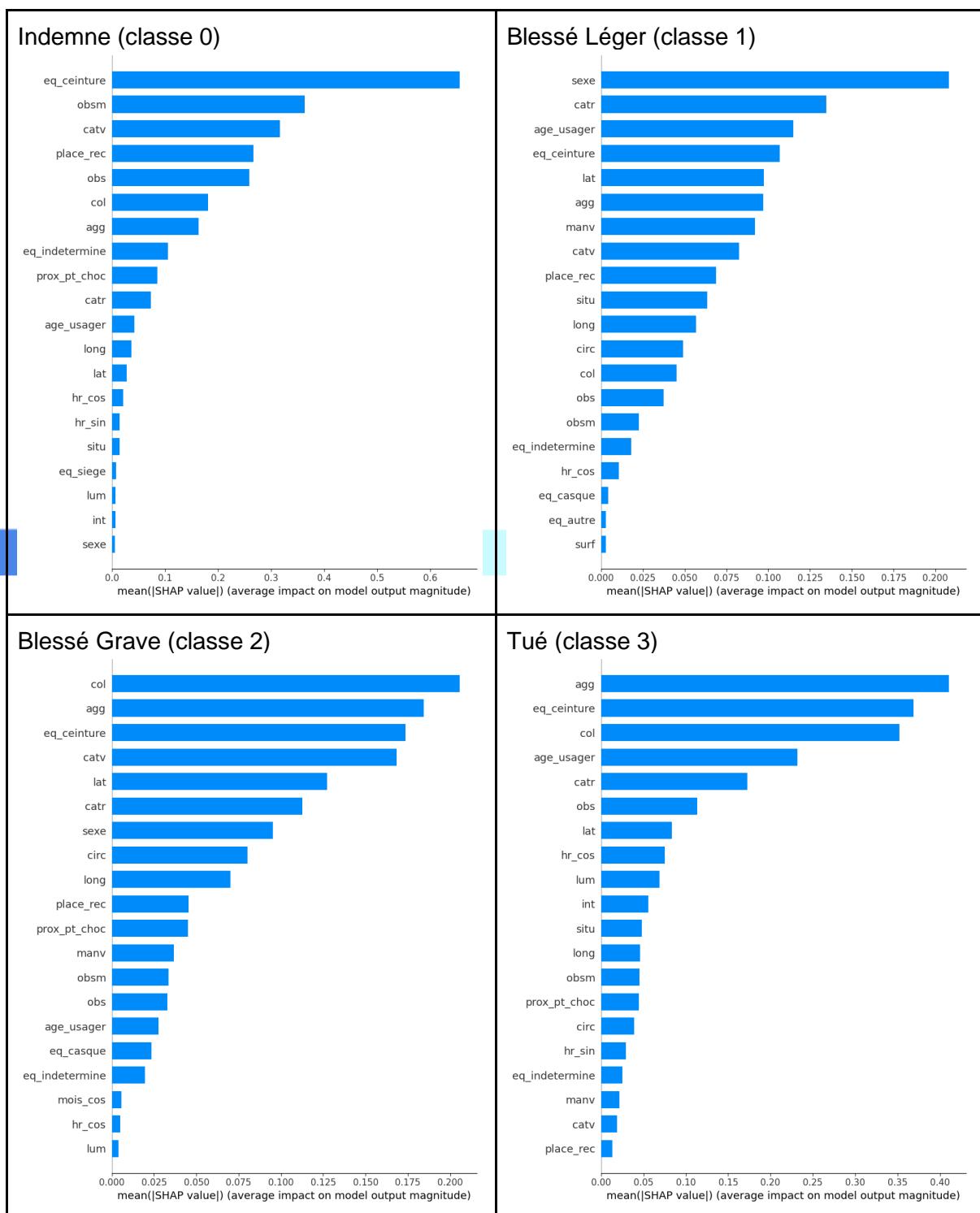


Figure 84 : Graphiques d'importance des variables selon SHAP

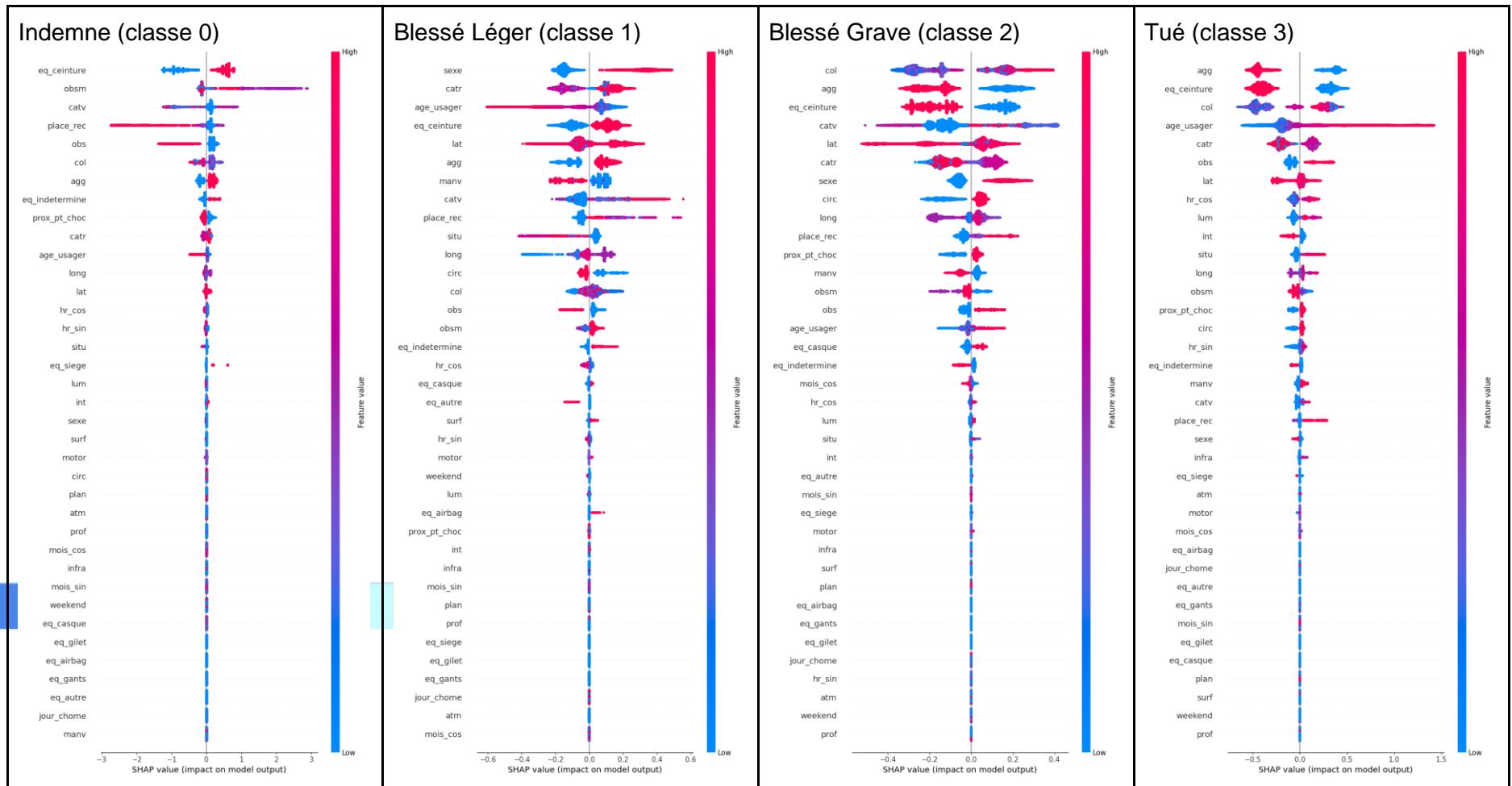


Figure 85 : Graphiques de densité des valeurs de SHAP

III.6 K-nearest neighbors (KNN)

III.6.1 Modèle de référence

A. Description du modèle

L'algorithme des K plus proches voisins (K-nearest neighbors, KNN) est un algorithme d'apprentissage **supervisé non paramétrique**, qui utilise la proximité des observations entre elles pour effectuer des classifications ou des prédictions. L'idée sous-jacente est que des points similaires peuvent être trouvés les uns à côté des autres : “*Dis-moi qui sont tes voisins, je te dirai qui tu es*”.

Pour les problèmes de classification, un vote aura lieu : le libellé de classe le plus fréquemment représenté parmi les K plus proches voisins du point à classifier lui sera attribué. Les K plus proches voisins peuvent avoir un poids égal dans le vote, ou un poids inversement proportionnel à leur distance du point à prédire.

A noter :

- 1) Les effectifs des classes étant déséquilibrés, nous avons travaillé sur un **sous-échantillon équilibré** de notre jeu de données, obtenu avec la fonction `RandomUnderSampler()`. L'échantillon ainsi obtenu contenait **9761 observations pour chaque classe de gravité** (indemnés, tués, blessés, blessés hospitalisés).
- 2) **Le calcul de la distance dans le modèle KNN nécessite que toutes les variables soient sur la même échelle de valeurs**, pour ne pas donner plus de poids a priori à certaines d'entre elles du simple fait de leur plage de valeur élevée. C'est pourquoi nous avons normalisé nos variables explicatives comme décrit plus haut dans la section sur la régression logistique.

B. Paramètres par défaut du modèle

Nous avons créé une première instance du modèle KNN avec les paramètres par défaut, c'est-à-dire notamment 5 voisins et la mesure de distance de Minkowski de puissance p=1 (équivalent à la distance de Manhattan).

Le tableau des performances de ce modèle obtenu avec `classification_report_imbalanced` est présenté ci-dessous :

Classe	pre	rec	spe	f1	geo	iba	sup
indemnés	0.69	0.73	0.77	0.71	0.75	0.56	36910
blessés	0.61	0.43	0.81	0.51	0.59	0.34	36078
blessés hospitalisés	0.34	0.37	0.87	0.35	0.56	0.30	14000
tués	0.12	0.46	0.91	0.19	0.64	0.39	2440
avg / total	0.59	0.55	0.80	0.56	0.65	0.43	89428

Tableau 12 : Rapport de classification

On retient qu'avec cette paramétrisation "par défaut", le f1-score est de 0.56, et que la prédiction de la classe « tués » est celle qui a la plus mauvaise précision (prec=0.12).

L'affichage de la matrice de confusion, ci-dessous, montre en effet que la majorité des personnes classées comme tuées appartiennent en réalité aux autres classes, tout simplement du fait de la prédominance de ces autres classes dans l'échantillon de test. Ainsi, un taux de mauvaise classification même faible sur ces autres classes va ajouter des faux positifs dans la catégorie « tués », en nombre largement plus important que le nombre de vrais positifs de cette classe.

Classe réelle / prédite	indemnes	blessés légers	blessés hospitalisés	tués
indemnes	26983	5905	2318	1704
blessés légers	10361	15677	7031	3009
blessés hospitalisés	1626	3781	5131	3462
tués	209	379	740	1112

Tableau 13 : Matrice de confusion de l'algorithme KNN (paramètres par défaut)

En normalisant la matrice de confusion par ligne, on voit mieux que les difficultés de classification ont lieu pour les classes du milieu : blessés légers et blessés hospitalisés (rappel < 0.5), que l'algorithme a du mal à discriminer des classes de gravité adjacentes. Ainsi, il classe plus souvent les blessés légers en indemnes ou en blessés hospitalisés que dans la bonne catégorie ($[0.29 + 0.19] > 0.43$) ; et les blessés hospitalisés en blessés légers ou tués ($[0.27 + 0.25] > 0.37$). Par contre, l'algorithme KNN classe peu les blessés légers en tués (8%) ou les blessés hospitalisés en indemnes (12%). Sa difficulté est donc vraiment de **bien prédire la finesse du continuum de gravité**.

Classe réelle / prédite	indemnes	blessés légers	blessés hospitalisés	tués
indemnes	0.73	0.16	0.06	0.05
blessés légers	0.29	0.43	0.19	0.08
blessés hospitalisés	0.12	0.27	0.37	0.25
tués	0.09	0.16	0.30	0.46

Tableau 14 : Matrice de confusion normalisée par ligne

III.6.2 Recherche de paramètres optimaux avec GridSearchCV

Nous avons ensuite cherché à optimiser notre algorithme de KNN en explorant avec GridSearchCV la grille construite sur le croisement suivant de paramètres :

- le paramètre de poids “weights”, a été fixé à “uniform” ou “distance”. Cette dernière valeur permet de pondérer la contribution des K plus proches voisins dans le vote en fonction de leur proximité au point que l’on cherche à classifier,
- le nombre de voisins a été étudié sur l’intervalle 5-30, par pas de 5,
- la puissance de la mesure de Minkowski a été fixée à 1 ou 2.

```
params = {'n_neighbors':[i for i in range(5,30,5)],
          'weights' : ['uniform','distance'],
          "p" : [i for i in range(1, 2)]}
```

Nous avons effectué une validation croisée à 5 volets stratifiés.

Le meilleur modèle était obtenu avec les paramètres suivants : {'n_neighbors': 25, 'p': 1, 'weights': 'distance'}. Voici ses performances :

Classe	pre	rec	spe	f1	geo	iba	sup
indemnes	0.70	0.78	0.77	0.74	0.77	0.60	36910
blessés	0.68	0.42	0.86	0.52	0.60	0.35	36078
blessés hospitalisés	0.36	0.36	0.88	0.36	0.57	0.30	14000
tués	0.12	0.58	0.88	0.20	0.72	0.50	2440
avg / total	0.62	0.57	0.83	0.58	0.67	0.45	89428

Tableau 15 : Rapport de classification pour le modèle optimisé

On note une amélioration de 0.02 points du f1-score (0.58 vs 0.56), avec une augmentation de 0.01 point (resp. 0.03 points) et du taux de rappel de la classe des tués : 0.58 vs 0.46, par rapport au modèle avec le paramétrage par défaut.

Malgré cette légère amélioration des performances, le problème reste toujours de distinguer les blessés légers et blessés hospitalisés des classes adjacentes, avec des rappels quasi-inchangés pour ces deux classes et une majorité des mauvais classements dans les classes de gravité adjacentes.

Classe réelle / prédite	indemnes	blessés	blessés hospitalisés	tués
indemnes	0.78	0.11	0.05	0.06
blessés	0.29	0.42	0.18	0.10
blessés hospitalisés	0.10	0.22	0.36	0.32
tués	0.07	0.09	0.25	0.58

Tableau 16 : Matrice de confusion normalisée par ligne pour le modèle optimisé

III.6.3 Sélection des features

L'algorithme du KNN “n'apprend pas” au fil des itérations à ignorer une feature qui n'apporte pas d'information par rapport à la variable cible, contrairement aux méthodes paramétriques comme les réseaux de neurones et la régression logistique. Dans le cas extrême, la présence de variables n'apportant que du bruit pourrait dégrader les performances de l'algorithme.

Nous avons décidé de ne conserver que les 20 variables les plus importantes (sur 34) telles que données par le meilleur modèle de Random Forest (voir plus haut) : eq_ceinture, age_usager, col, agg, lat, place_rec, obsm, obs, catv, long, eq_casque, catr, situ, heure, circ, prox_pt_choc, mois, sexe, lum, motor.

Le modèle KNN est toujours le même : 25 voisins et distance de Minkowski de puissance 1.

Les performances sont les suivantes, c'est-à-dire quasiment les mêmes que lorsqu'on utilise les 34 variables :

Classe	pre	rec	spe	f1	geo	iba	sup
indemnes	0.71	0.76	0.78	0.74	0.77	0.60	36910
blessés	0.67	0.44	0.85	0.53	0.61	0.36	36078
blessés hospitalisés	0.36	0.36	0.88	0.36	0.56	0.30	14000
tués	0.12	0.59	0.88	0.20	0.72	0.50	2440
avg / total	0.62	0.56	0.83	0.58	0.67	0.45	89428

Tableau 17 : Rapport de classification avec sélection de variables

Ce modèle mettant 32.2 secondes à tourner au lieu de 53.7 secondes pour le modèle précédent, on a tout intérêt à supprimer les variables peu informatives avant de lancer le KNN.

Certains chercheurs (notamment (Bhardwaj, Mishra, & Desikan, s.d.) proposent d'aller plus loin en appliquant en préalable du KNN une procédure de scaling qui tient compte de l'importance des variables telle que déterminée dans une random forest. Ainsi, la distance entre deux points sera “tirée” par les variables les plus liées à la réponse, et l'on pourrait ainsi améliorer les performances de classification du KNN, peut-être notamment sur sa capacité à discriminer les blessés légers des indemnes et des blessés hospitalisés, et les blessés hospitalisés des blessés légers et des tués. Faute de temps, cette option n'a pas pu être explorée.

IV ESSAIS D'OPTIMISATION DE LA PRÉDICTION DE LA GRAVITÉ DE L'ACCIDENT – MACHINE LEARNING

IV.1 Classification binaire

En réalisant la classification multi-classes, nous nous sommes confrontés au problème de prédiction des personnes tuées (classe 2) et des blessés hospitalisés (classe 3) qui sont respectivement des classes très fortement et fortement minoritaires. Nous avons donc décidé de tester une classification binaire en créant 4 jeux de données à partir de notre jeu de données initial : ‘indemnés’ vs ‘autres’, ‘tués’ vs ‘autres’, ‘blessés hospitalisés’ vs ‘autres’, ‘blessés légers’ vs ‘autres’.

IV.1.1 Modélisation

A. Paramétrage du modèle

Les meilleurs résultats ayant été obtenus avec un algorithme Random Forest, nous décidons d’appliquer le même algorithme en recherchant, de la même manière que décrite précédemment, les paramètres optimaux.

Puis, nous appliquons un nouveau paramètre (possible uniquement sur la classification binaire) : `monotonic_cts`. Pour chaque variable, nous testons les 3 valeurs de `monotonic_cts` possibles (-1, 0 et 1) et nous enregistrons les résultats dans un tableau que l’on trie par f1-score-macro décroissant. Si le f1-score-macro est amélioré alors on garde cette valeur de `monotonic_cts` pour cette variable et on relance la recherche pour savoir s’il faut aussi utiliser une autre variable dans `monotonic_cts`. En revanche, si le f1 macro est moins bon, on ne garde pas cette valeur.

B. Résultats

Pour les indemnés (Train accuracy = 85.42% - Test accuracy = 80.03%)																																														
Paramètres :																																														
{‘bootstrap’: True, ‘class_weight’: ‘balanced’, ‘criterion’: ‘entropy’, ‘max_depth’: 20, ‘min_samples_leaf’: 1, ‘min_samples_split’: 10, ‘n_estimators’: 100, ‘n_jobs’: -1, ‘random_state’: 42}																																														
C. ‘monotonic_cts’ à -1 pour ‘eq_gilet’ et à 0 pour toutes les autres variables																																														
<table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.88</td><td>0.77</td><td>0.82</td><td>65647</td></tr><tr><td>1</td><td>0.72</td><td>0.85</td><td>0.78</td><td>46137</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.80</td><td>111784</td></tr><tr><td>macro avg</td><td>0.80</td><td>0.81</td><td>0.80</td><td>111784</td></tr><tr><td>weighted avg</td><td>0.81</td><td>0.80</td><td>0.80</td><td>111784</td></tr></tbody></table>			precision	recall	f1-score	support	0	0.88	0.77	0.82	65647	1	0.72	0.85	0.78	46137	accuracy			0.80	111784	macro avg	0.80	0.81	0.80	111784	weighted avg	0.81	0.80	0.80	111784	<table><thead><tr><th>Classes prédictes</th><th>0</th><th>1</th></tr><tr><th>Classes réelles</th><th></th><th></th></tr></thead><tbody><tr><td>0</td><td>50565</td><td>15082</td></tr><tr><td>1</td><td>7137</td><td>39000</td></tr></tbody></table>			Classes prédictes	0	1	Classes réelles			0	50565	15082	1	7137	39000
	precision	recall	f1-score	support																																										
0	0.88	0.77	0.82	65647																																										
1	0.72	0.85	0.78	46137																																										
accuracy			0.80	111784																																										
macro avg	0.80	0.81	0.80	111784																																										
weighted avg	0.81	0.80	0.80	111784																																										
Classes prédictes	0	1																																												
Classes réelles																																														
0	50565	15082																																												
1	7137	39000																																												
Pour les tués (Train accuracy = 95.58% - Test accuracy = 94.06%)																																														
Paramètres :																																														
{‘bootstrap’: True, ‘class_weight’: {0: 1, 1: 18}, ‘criterion’: ‘entropy’, ‘max_depth’: None, ‘min_samples_leaf’: 1, ‘min_samples_split’: 2, ‘n_estimators’: 100, ‘n_jobs’: -1, ‘random_state’: 42 }																																														
D. ‘monotonic_cts’ à -1 pour ‘circ’ et à 0 pour toutes les autres variables																																														

<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>0</td><td>0.98</td><td>0.96</td><td>0.97</td><td>108734</td></tr> <tr> <td>1</td><td>0.20</td><td>0.39</td><td>0.27</td><td>3050</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.94</td><td>111784</td></tr> <tr> <td>macro avg</td><td>0.59</td><td>0.67</td><td>0.62</td><td>111784</td></tr> <tr> <td>weighted avg</td><td>0.96</td><td>0.94</td><td>0.95</td><td>111784</td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.98	0.96	0.97	108734	1	0.20	0.39	0.27	3050	accuracy			0.94	111784	macro avg	0.59	0.67	0.62	111784	weighted avg	0.96	0.94	0.95	111784	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Classes prédictes</th><th>0</th><th>1</th></tr> </thead> <tbody> <tr> <td>Classes réelles</td><td></td><td></td></tr> <tr> <td>0</td><td>103942</td><td>4792</td></tr> <tr> <td>1</td><td>1849</td><td>1201</td></tr> </tbody> </table>	Classes prédictes	0	1	Classes réelles			0	103942	4792	1	1849	1201
	precision	recall	f1-score	support																																							
0	0.98	0.96	0.97	108734																																							
1	0.20	0.39	0.27	3050																																							
accuracy			0.94	111784																																							
macro avg	0.59	0.67	0.62	111784																																							
weighted avg	0.96	0.94	0.95	111784																																							
Classes prédictes	0	1																																									
Classes réelles																																											
0	103942	4792																																									
1	1849	1201																																									
Pour les blessés hospitalisés (Train accuracy = 85.6% - Test accuracy = 82.96%)																																											
Paramètres : <pre>{'bootstrap': True, 'class_weight': {0: 1, 1: 3}, 'criterion': 'entropy', 'max_depth': 14, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100, 'n_jobs': -1, 'random_state': 42}</pre> E. 'monotonic_cts' à 1 pour 'int', à -1 pour 'jour_chome' et à 0 pour toutes les autres variables																																											
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0.92</td> <td>0.87</td> <td>0.90</td> <td>94284</td> </tr> <tr> <td>1</td> <td>0.46</td> <td>0.59</td> <td>0.52</td> <td>17500</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.83</td> <td>111784</td> </tr> <tr> <td>macro avg</td> <td>0.69</td> <td>0.73</td> <td>0.71</td> <td>111784</td> </tr> <tr> <td>weighted avg</td> <td>0.85</td> <td>0.83</td> <td>0.84</td> <td>111784</td> </tr> </tbody> </table>		precision	recall	f1-score	support	0	0.92	0.87	0.90	94284	1	0.46	0.59	0.52	17500	accuracy			0.83	111784	macro avg	0.69	0.73	0.71	111784	weighted avg	0.85	0.83	0.84	111784	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Classes prédictes</th> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>Classes réelles</td> <td></td> <td></td> </tr> <tr> <td>0</td> <td>82473</td> <td>11811</td> </tr> <tr> <td>1</td> <td>7242</td> <td>10258</td> </tr> </tbody> </table>	Classes prédictes	0	1	Classes réelles			0	82473	11811	1	7242	10258
	precision	recall	f1-score	support																																							
0	0.92	0.87	0.90	94284																																							
1	0.46	0.59	0.52	17500																																							
accuracy			0.83	111784																																							
macro avg	0.69	0.73	0.71	111784																																							
weighted avg	0.85	0.83	0.84	111784																																							
Classes prédictes	0	1																																									
Classes réelles																																											
0	82473	11811																																									
1	7242	10258																																									
Pour les blessés légers (Train accuracy = 80.54% - Test accuracy = 72.78%)																																											
Paramètres : <pre>{'bootstrap': False, 'class_weight': {0: 3, 1: 4}, 'criterion': 'gini', 'max_depth': 15, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100, 'n_jobs': -1, 'random_state': 42}</pre> F. 'monotonic_cts' à -1 pour 'motor' et à 0 pour toutes les autres variables																																											
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0.76</td> <td>0.79</td> <td>0.78</td> <td>66687</td> </tr> <tr> <td>1</td> <td>0.67</td> <td>0.64</td> <td>0.65</td> <td>45097</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.73</td> <td>111784</td> </tr> <tr> <td>macro avg</td> <td>0.72</td> <td>0.71</td> <td>0.71</td> <td>111784</td> </tr> <tr> <td>weighted avg</td> <td>0.73</td> <td>0.73</td> <td>0.73</td> <td>111784</td> </tr> </tbody> </table>		precision	recall	f1-score	support	0	0.76	0.79	0.78	66687	1	0.67	0.64	0.65	45097	accuracy			0.73	111784	macro avg	0.72	0.71	0.71	111784	weighted avg	0.73	0.73	0.73	111784	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Classes prédictes</th> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>Classes réelles</td> <td></td> <td></td> </tr> <tr> <td>0</td> <td>52706</td> <td>13981</td> </tr> <tr> <td>1</td> <td>16445</td> <td>28652</td> </tr> </tbody> </table>	Classes prédictes	0	1	Classes réelles			0	52706	13981	1	16445	28652
	precision	recall	f1-score	support																																							
0	0.76	0.79	0.78	66687																																							
1	0.67	0.64	0.65	45097																																							
accuracy			0.73	111784																																							
macro avg	0.72	0.71	0.71	111784																																							
weighted avg	0.73	0.73	0.73	111784																																							
Classes prédictes	0	1																																									
Classes réelles																																											
0	52706	13981																																									
1	16445	28652																																									

Figure 86 : Métriques et matrice de confusion selon le jeu de données binaire utilisé

La classification binaire connaît les mêmes difficultés que la classification multiclass pour prédire les classes minoritaires (les tués et les blessés hospitalisés). Cependant, l'accuracy est très nettement améliorée.

IV.1.2 Comparaison des résultats avec la classification multiclass

Pour permettre la comparaison, nous allons diviser la matrice de confusion obtenue lors de la classification multi-classes de sorte à obtenir des résultats sous forme de classification binaire (Figure 87) :

La classification binaire permet d'améliorer de presque 2%, dans tous les cas, le nombre de vrais positifs :

- 'indemnes' vs 'autres' : diminution de 232 'indemnes' vrais positifs, mais augmentation de 2108 'autres' vrais positifs,
- 'tués' vs 'autres' : diminution de 287 'tués' vrais positifs, mais augmentation de 2323 'autres' vrais positifs

- ‘blessés hospitalisés’ vs ‘autres’ : augmentation de 1626 ‘blessés hospitalisés’ vrais positifs et de 602 ‘autres’ vrais positifs,
- ‘blessés légers’ vs ‘autres’ ; augmentation de 9565 ‘blessés légers’ vrais positifs, mais diminution de 7354 ‘autres’ vrais positifs.

La classification binaire est donc meilleure que la classification multi-classes.

IV.1.3 Interprétabilité des résultats

A. Analyse de l’importance donnée par le meilleur modèle

Nous nous sommes intéressés à l’importance donnée à chaque variable par l’algorithme pour chaque jeu de données binaire (Figure 88).

Nous pouvons constater que, dans les 3 premières variables, on retrouve systématiquement la variable correspondant à l'**utilisation ou non de la ceinture de sécurité** qui était d’ailleurs la variable ayant le plus d’importance lors de la classification multi-classes.

De plus, les variables correspondant au **type de collision**, à la **latitude**, à la **longitude**, à l'**obstacle mobile heurté** sont présentes dans 3 jeux de données sur 4.

B. Analyse de l’interprétation du meilleur modèle avec SHAP

En étudiant les graphiques d’importance des variables (Figure 89), mais également les graphiques de densité des valeurs SHAP pour chaque variable (Figure 90) en fonction de la modalité de la variable cible ‘grav’, nous pouvons dire que les 5 paramètres qui influent positivement sur la classification sont :

- **pour les indemnes :**
 - l'**utilisation de la ceinture de sécurité**,
 - **être à l’avant du véhicule** (les valeurs les plus faibles de la variable catégorielle ‘place_rec’ sont soit conducteur et/ou passager avant),
 - **ne pas porter de casque** (certainement pour le fait de ne pas être en vélo/trottinette ou moto),
 - **ne pas heurter d’obstacle fixe**.

Pour la variable ‘catv’ (variable catégorielle regroupant 6 modalités), la distinction entre les valeurs sur le graphique de densité des valeurs est plus floue et ne permet pas d’établir formellement quelles modalités influent positivement sur la classification.

- **pour les tués :**
 - **ne pas utiliser de ceinture de sécurité**,
 - **rouler hors d’agglomération**,
 - **un âge plus avancé**.

Pour les variables ‘catr’ (variable catégorielle regroupant 8 modalités) et ‘col’ (variable catégorielle regroupant 7 modalités) la distinction entre les valeurs sur le graphique de densité des valeurs est plus floue et ne permet pas d’établir formellement quelles modalités influent positivement sur la classification.

- **pour les blessés hospitalisés :**
 - **ne pas utiliser de ceinture de sécurité**,
 - **rouler hors d’agglomération**,
 - **rouler sur une route bidirectionnelle**,

Pour les variables 'lat' (variable numérique) et 'col' (variable catégorielle regroupant 7 modalités), la distinction entre les valeurs sur le graphique de densité des valeurs est plus floue et ne permet pas d'établir formellement quelles modalités ou valeurs influent positivement sur la classification.

- **pour les blessés légers :**
 - **être à l'arrière du véhicule ou piéton**(les valeurs les plus élevées de la variable catégorielle 'place_rec' sont soit passager arrière, soit piéton),
 - **être une femme,**
 - **ne pas utiliser de ceinture de sécurité,**
 - **porter un casque** (certainement pour le fait d'être en vélo/trottinette ou moto),

Pour la variable 'catv' (variable catégorielle regroupant 6 modalités), la distinction entre les valeurs sur le graphique de densité des valeurs est plus floue et ne permet pas d'établir formellement quelles modalités influent positivement sur la classification.

On peut noter une légère différence de l'influence des variables par rapport à la classification multi-classes.



Figure 87 : Réarrangement de la matrice de confusion pour de la classification multiclass et comparaison avec la classification binaire

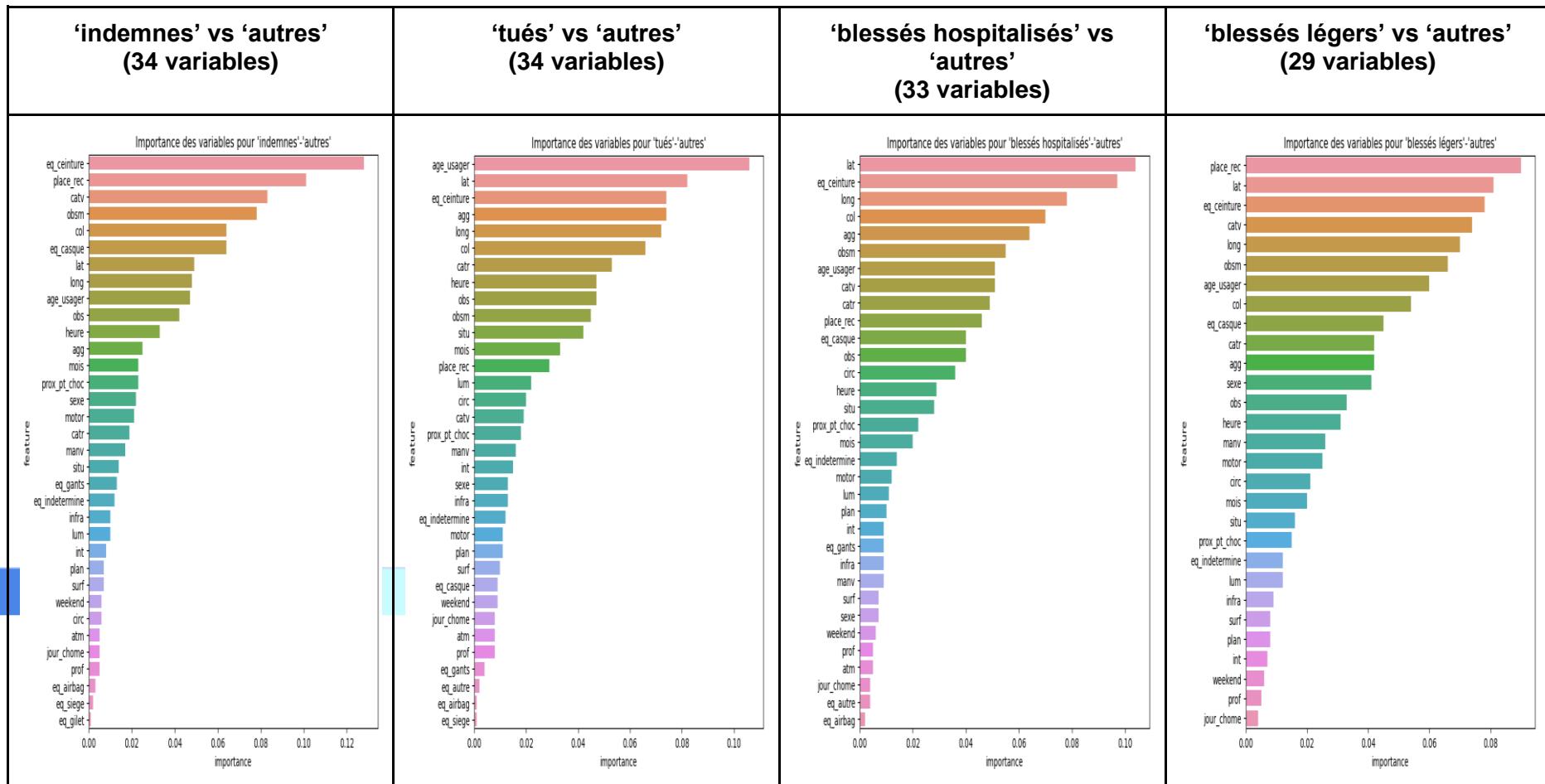


Figure 88 : Importance des variables selon le jeu de données binaires

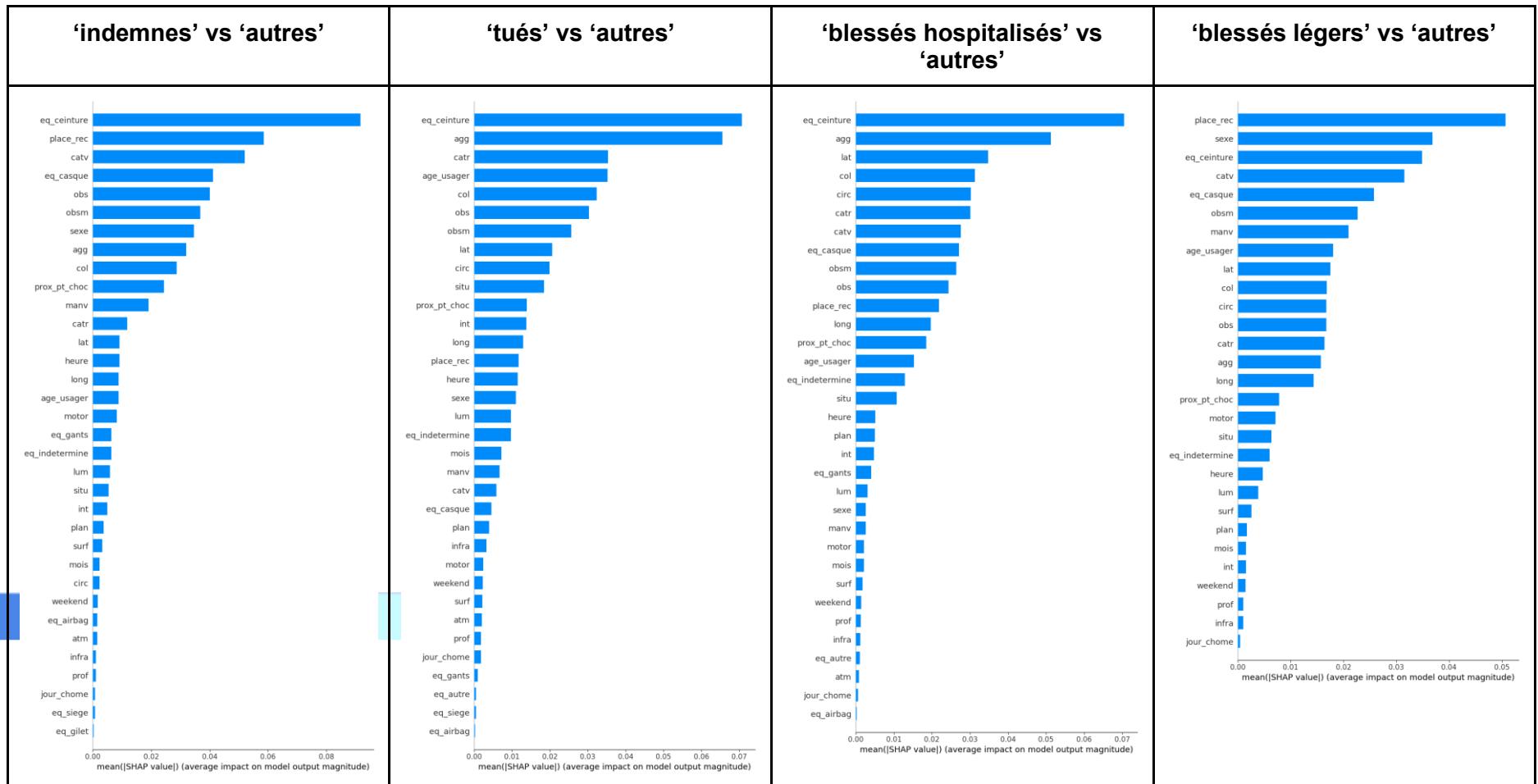
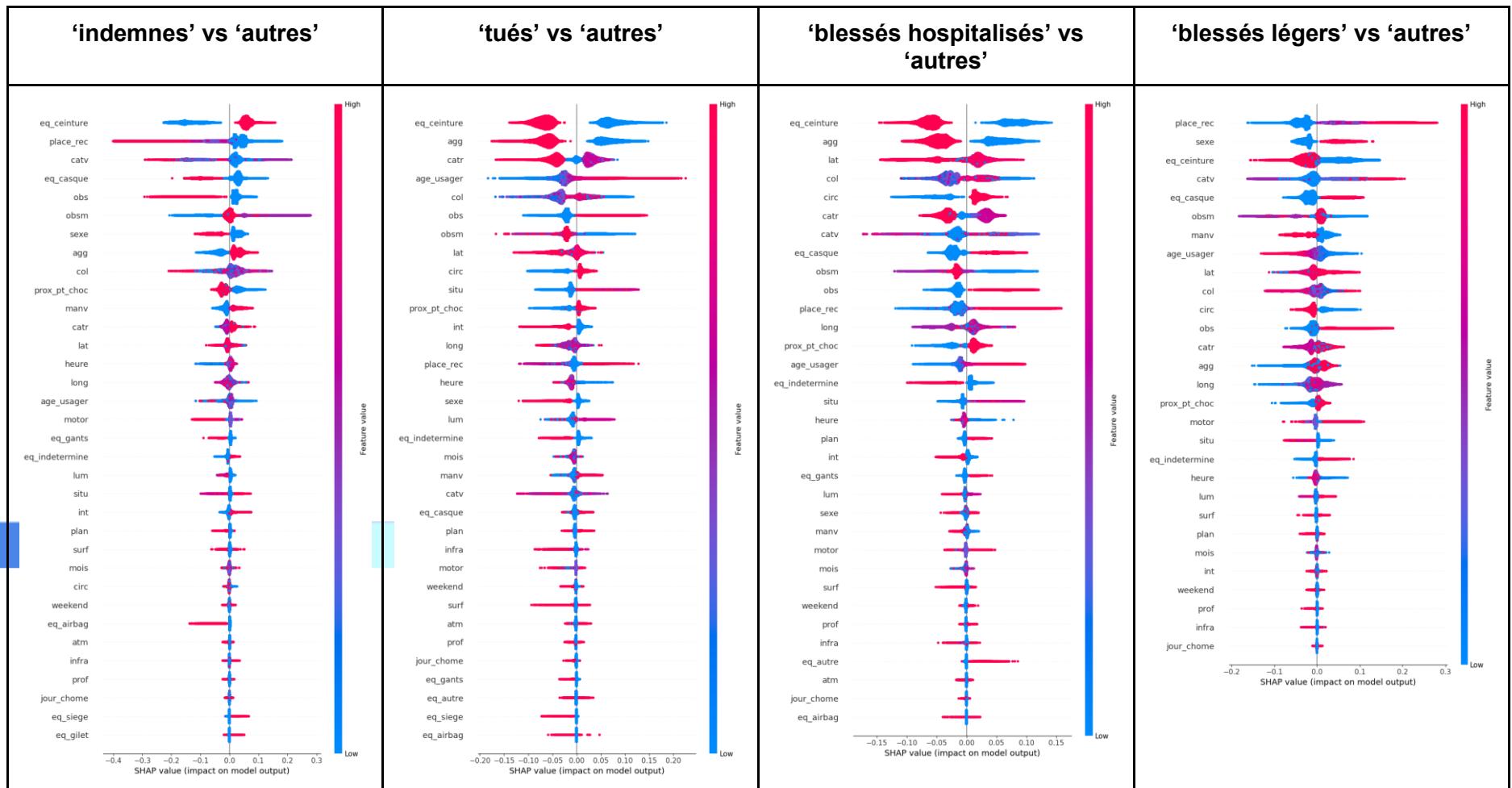


Figure 89 : Graphiques d'importance des variables selon SHAP selon le jeu de données binaires



IV.2 Etude des variables 'catv', 'obs' et 'obsm'

Nous nous sommes aussi intéressés à l'amélioration des résultats par modification du jeu de données initial. Pour cela, nous avons regardé plus précisément l'influence des variables 'catv', 'obs' et 'obsm'.

IV.2.1 Modifications des variables

A. Modification de la variable 'catv' et création de 'catv_percute'

La variable 'catv', qui prend comme modalité les différents types de véhicule, comporte 2 informations :

- le type de véhicule dans lequel la personne accidentée se situe quand elle est conducteur ou passager,
- le type de véhicule qui a percuté la personne accidentée quand elle est piétonne.

Pour rendre compte de cela, nous décidons de créer la variable 'catv_percute' qui indique le véhicule qui a percuté la personne accidentée. Les modalités de 'catv_percute' pour les piétons (ayant la 'place_rec' = 4) sont les mêmes que celles de 'catv'. Par contre pour les autres modalités de 'place_rec' correspondant au conducteurs et passagers, on attribue une nouvelle modalité '6' car on ne connaît pas le véhicule qui a percuté :

- 0 - Voiture
- 1 - Moto
- 2 - Poid lourd
- 3 - Transport en commun
- 4 - Vélo/Trottinette
- 5 - Autre véhicule
- 6 - Véhicule inconnu

Un nouvelle modalité '6' est créée dans la variable 'catv' et est attribuée à tous les piétons :

- 0 - Voiture
- 1 - Moto
- 2 - Poid lourd
- 3 - Transport en commun
- 4 - Vélo/Trottinette
- 5 - Autre véhicule
- 6 - Piéton

B. Modification des variables 'obs' et 'obsm'

Les variables 'obs' et 'obsm' correspondent respectivement à l'obstacle fixe ou mobile heurté par le véhicule. Or un piéton, par définition, n'est pas un véhicule et ne va donc pas heurter d'objet. Cependant, lorsque l'on regarde les valeurs de 'obs' et 'obsm' pour les piétons ('place_rec' = 4), on se rend compte que certaines valeurs sont différentes de 0. Étant donné que, pour les piétons, la case 'catv' a été remplie par rapport au véhicule qui a percuté le piéton, il se peut que ces cases l'aient été de la même manière. Nous décidons donc de mettre les valeurs de 'obs' et 'obsm' à 0 lorsqu'il s'agit d'un piéton.

IV.2.2 Modélisation

Nous reprenons le meilleur modèle que nous avons obtenu pour la classification multiconfondue : un algorithme de Random forest.

IV.2.3 Résultats et comparaison avec le jeu de données initial

La comparaison avec le jeu de données initiale (Figure 91) nous montre que ces modifications permettent d'obtenir un meilleur classement des personnes accidentées :

Jeu de données initial	Jeu de données avec modification de 'catv', 'obs', 'obsm' + création de 'catv_percute'																																																																																
Random Forest (Train accuracy = 65.52%) (Test accuracy = 61.22%)	Random Forest (Train accuracy = 65.83%) (Test accuracy = 61.36%)																																																																																
paramètres : {'bootstrap': True, 'class_weight': 'balanced', 'criterion': 'entropy', 'max_depth': 13, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100, 'n_jobs': -1, 'random_state': 42}	paramètres : {'bootstrap': True, 'class_weight': 'balanced', 'criterion': 'entropy', 'max_depth': 13, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100, 'n_jobs': -1, 'random_state': 42}																																																																																
<table> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0.70</td> <td>0.85</td> <td>0.77</td> <td>46137</td> </tr> <tr> <td>2</td> <td>0.17</td> <td>0.49</td> <td>0.26</td> <td>3050</td> </tr> <tr> <td>3</td> <td>0.41</td> <td>0.49</td> <td>0.45</td> <td>17500</td> </tr> <tr> <td>4</td> <td>0.74</td> <td>0.42</td> <td>0.54</td> <td>45097</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.61</td> <td>111784</td> </tr> <tr> <td>macro avg</td> <td>0.51</td> <td>0.56</td> <td>0.50</td> <td>111784</td> </tr> <tr> <td>weighted avg</td> <td>0.66</td> <td>0.61</td> <td>0.61</td> <td>111784</td> </tr> </tbody> </table>		precision	recall	f1-score	support	1	0.70	0.85	0.77	46137	2	0.17	0.49	0.26	3050	3	0.41	0.49	0.45	17500	4	0.74	0.42	0.54	45097	accuracy			0.61	111784	macro avg	0.51	0.56	0.50	111784	weighted avg	0.66	0.61	0.61	111784	<table> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0.70</td> <td>0.85</td> <td>0.77</td> <td>46137</td> </tr> <tr> <td>2</td> <td>0.18</td> <td>0.49</td> <td>0.26</td> <td>3050</td> </tr> <tr> <td>3</td> <td>0.41</td> <td>0.50</td> <td>0.45</td> <td>17500</td> </tr> <tr> <td>4</td> <td>0.74</td> <td>0.43</td> <td>0.54</td> <td>45097</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.61</td> <td>111784</td> </tr> <tr> <td>macro avg</td> <td>0.51</td> <td>0.57</td> <td>0.50</td> <td>111784</td> </tr> <tr> <td>weighted avg</td> <td>0.66</td> <td>0.61</td> <td>0.61</td> <td>111784</td> </tr> </tbody> </table>		precision	recall	f1-score	support	1	0.70	0.85	0.77	46137	2	0.18	0.49	0.26	3050	3	0.41	0.50	0.45	17500	4	0.74	0.43	0.54	45097	accuracy			0.61	111784	macro avg	0.51	0.57	0.50	111784	weighted avg	0.66	0.61	0.61	111784
	precision	recall	f1-score	support																																																																													
1	0.70	0.85	0.77	46137																																																																													
2	0.17	0.49	0.26	3050																																																																													
3	0.41	0.49	0.45	17500																																																																													
4	0.74	0.42	0.54	45097																																																																													
accuracy			0.61	111784																																																																													
macro avg	0.51	0.56	0.50	111784																																																																													
weighted avg	0.66	0.61	0.61	111784																																																																													
	precision	recall	f1-score	support																																																																													
1	0.70	0.85	0.77	46137																																																																													
2	0.18	0.49	0.26	3050																																																																													
3	0.41	0.50	0.45	17500																																																																													
4	0.74	0.43	0.54	45097																																																																													
accuracy			0.61	111784																																																																													
macro avg	0.51	0.57	0.50	111784																																																																													
weighted avg	0.66	0.61	0.61	111784																																																																													
<table> <thead> <tr> <th>Classes prédites</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> </tr> <tr> <th>Classes réelles</th> <th></th> <th></th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>39232</td> <td>1262</td> <td>2339</td> <td>3304</td> </tr> <tr> <td>2</td> <td>288</td> <td>1488</td> <td>1016</td> <td>258</td> </tr> <tr> <td>3</td> <td>2225</td> <td>3578</td> <td>8632</td> <td>3065</td> </tr> <tr> <td>4</td> <td>14677</td> <td>2275</td> <td>9058</td> <td>19087</td> </tr> </tbody> </table>	Classes prédites	1	2	3	4	Classes réelles					1	39232	1262	2339	3304	2	288	1488	1016	258	3	2225	3578	8632	3065	4	14677	2275	9058	19087	<table> <thead> <tr> <th>Classes prédites</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> </tr> <tr> <th>Classes réelles</th> <th></th> <th></th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>39227</td> <td>1108</td> <td>2415</td> <td>3387</td> </tr> <tr> <td>2</td> <td>290</td> <td>1483</td> <td>1020</td> <td>257</td> </tr> <tr> <td>3</td> <td>2205</td> <td>3503</td> <td>8716</td> <td>3076</td> </tr> <tr> <td>4</td> <td>14445</td> <td>2089</td> <td>9368</td> <td>19195</td> </tr> </tbody> </table>	Classes prédites	1	2	3	4	Classes réelles					1	39227	1108	2415	3387	2	290	1483	1020	257	3	2205	3503	8716	3076	4	14445	2089	9368	19195																				
Classes prédites	1	2	3	4																																																																													
Classes réelles																																																																																	
1	39232	1262	2339	3304																																																																													
2	288	1488	1016	258																																																																													
3	2225	3578	8632	3065																																																																													
4	14677	2275	9058	19087																																																																													
Classes prédites	1	2	3	4																																																																													
Classes réelles																																																																																	
1	39227	1108	2415	3387																																																																													
2	290	1483	1020	257																																																																													
3	2205	3503	8716	3076																																																																													
4	14445	2089	9368	19195																																																																													
61.22% de vraies positifs	61.36% de vraies positifs																																																																																

Figure 91 : Comparaison des métriques et matrice de confusion pour le jeu de données initial et le jeu de données avec modification de variables

En comparant les 2 modèles, on peut voir que le retraitement des variables 'catv', 'obs' et 'obsm' ne change pas le f1_score des différentes modalités, mais permet de légèrement améliorer l'accuracy. De plus, on peut voir que le nombre de vrai positif :

- diminue de 5 pour les indemnes,
- diminue de 5 pour les tués,
- augmente de 84 pour les blessés hospitalisés,
- augmente de 108 pour les blessés légers.

L'ajout de la variable 'catv_percute' pour les piétons (qui représente seulement 7.62% du jeu de données) a permis d'améliorer les résultats de 0.14%. Nous ne disposons pas de cette information lorsque la personne accidentée est conducteur ou passager. Or le type de véhicule qui percute peut forcément impacter la gravité de l'accident. Il serait donc intéressant d'ajouter cette variable lors de la collecte des informations à la suite d'un accident afin d'avoir la variable 'catv_percute' remplie pour tous les accidentés et non pas seulement pour les piétons.

IV.3 Conclusion des modélisations de Machine Learning

Pour répondre à la problématique de mieux comprendre la gravité des blessures causées par les accidents de la route en France, nous avons eu recours à des algorithmes de classification issus de différentes familles, à savoir :

- des algorithmes linéaires : régression logistique,
- des algorithmes de regroupement : machines à vecteurs de support (SVM), K plus proches voisins (KNN)
- des algorithmes à arbres de décision : arbre de décision, forêt aléatoire,
- des apprentissages d'ensemble : CatBoost, XGBoost

Pour optimiser les hyperparamètres de ces modèles, différentes méthodes d'exploration ont été mises en oeuvre, telles que la validation croisée (cross-validation), la recherche par grille (GridSearchCV), la recherche aléatoire (RandomSearch) ou l'optimisation bayésienne (Optuna).

De façon générale, les gains obtenus par ces procédures d'optimisation restent relativement modérés (Figure 92), ce qui peut être lié à l'absence dans la base de données de variables fortement influentes pour prédire la gravité des accidents routiers. Rappelons que des informations telles que les vitesses lors des accidents, l'état de santé des usagers, ... sont absentes (non communiquées) de la base de données mise à disposition par le gouvernement.

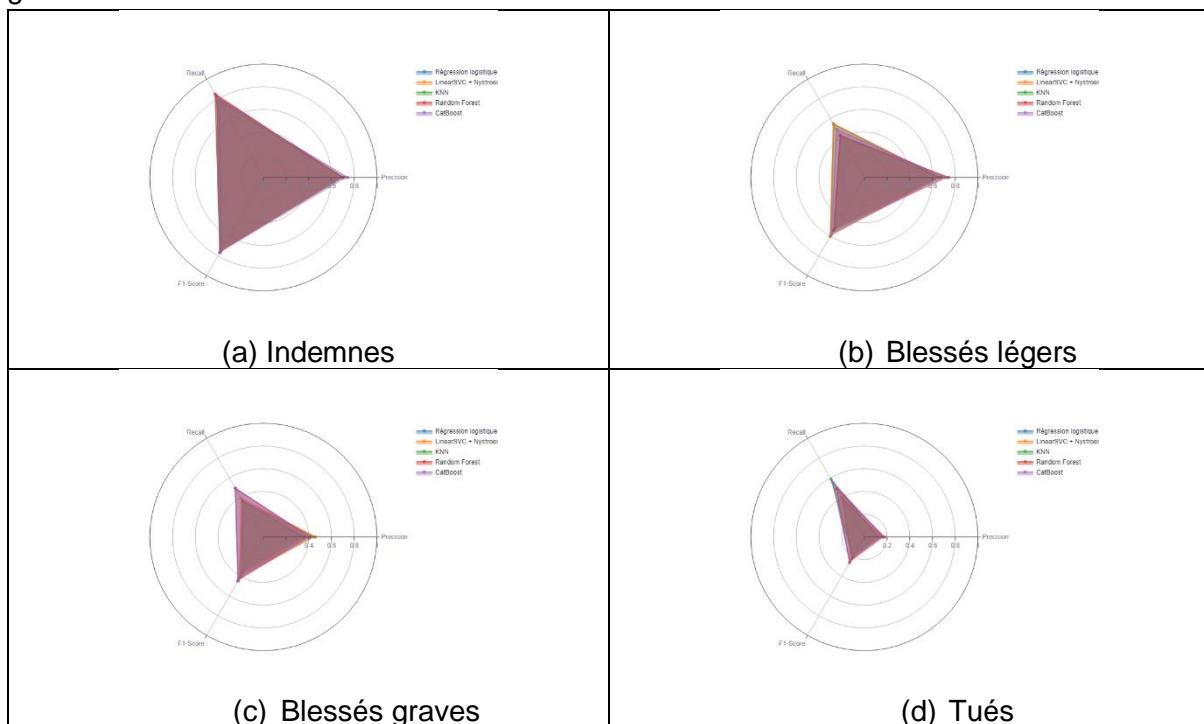


Figure 92 : Diagrammes radars de comparaison des modèles en termes de Precision, Recall, Specificity, F1-Score et Index Balanced Accuracy

Finalement, notre meilleur modèle repose sur les forêts aléatoires et atteint les performances listées à la Figure 93.

Random Forest																																																																											
(Train accuracy = 65.52% - Test accuracy = 61.22%)																																																																											
Paramètres :																																																																											
{'bootstrap': True, 'class_weight': 'balanced', 'criterion': 'entropy', 'max_depth': 13, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100, 'n_jobs': -1, 'random_state': 42}																																																																											
<table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>1</td><td>0.70</td><td>0.85</td><td>0.77</td><td>46137</td></tr> <tr> <td>2</td><td>0.17</td><td>0.49</td><td>0.26</td><td>3050</td></tr> <tr> <td>3</td><td>0.41</td><td>0.49</td><td>0.45</td><td>17500</td></tr> <tr> <td>4</td><td>0.74</td><td>0.42</td><td>0.54</td><td>45097</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.61</td><td>111784</td></tr> <tr> <td>macro avg</td><td>0.51</td><td>0.56</td><td>0.50</td><td>111784</td></tr> <tr> <td>weighted avg</td><td>0.66</td><td>0.61</td><td>0.61</td><td>111784</td></tr> </tbody> </table>				precision	recall	f1-score	support	1	0.70	0.85	0.77	46137	2	0.17	0.49	0.26	3050	3	0.41	0.49	0.45	17500	4	0.74	0.42	0.54	45097	accuracy			0.61	111784	macro avg	0.51	0.56	0.50	111784	weighted avg	0.66	0.61	0.61	111784	<table border="1"> <thead> <tr> <th>Clases prédictes</th><th>1</th><th>2</th><th>3</th><th>4</th></tr> <tr> <th>Clases réelles</th><th></th><th></th><th></th><th></th></tr> </thead> <tbody> <tr> <td>1</td><td>39232</td><td>1262</td><td>2339</td><td>3304</td></tr> <tr> <td>2</td><td>288</td><td>1488</td><td>1016</td><td>258</td></tr> <tr> <td>3</td><td>2225</td><td>3578</td><td>8632</td><td>3065</td></tr> <tr> <td>4</td><td>14677</td><td>2275</td><td>9058</td><td>19087</td></tr> </tbody> </table>			Clases prédictes	1	2	3	4	Clases réelles					1	39232	1262	2339	3304	2	288	1488	1016	258	3	2225	3578	8632	3065	4	14677	2275	9058	19087
	precision	recall	f1-score	support																																																																							
1	0.70	0.85	0.77	46137																																																																							
2	0.17	0.49	0.26	3050																																																																							
3	0.41	0.49	0.45	17500																																																																							
4	0.74	0.42	0.54	45097																																																																							
accuracy			0.61	111784																																																																							
macro avg	0.51	0.56	0.50	111784																																																																							
weighted avg	0.66	0.61	0.61	111784																																																																							
Clases prédictes	1	2	3	4																																																																							
Clases réelles																																																																											
1	39232	1262	2339	3304																																																																							
2	288	1488	1016	258																																																																							
3	2225	3578	8632	3065																																																																							
4	14677	2275	9058	19087																																																																							

Figure 93 : Métriques et matrice de confusion pour le meilleur modèle

Tous ces algorithmes se heurtent à la difficulté de gestion d'un jeu de données déséquilibré. La création d'un modèle pénalisé par l'ajout de poids différents aux différentes classes améliore les résultats (plus que les techniques de rééquilibrage du jeu de données), mais les résultats continuent de montrer une faiblesse dans la prédiction des classes minoritaires.

Les techniques d'interprétabilité des résultats des modèles de machine learning ont été appliquées pour mieux comprendre les facteurs contribuant aux accidents de la route. Des plus simples (ordres de grandeur des coefficients de régression logistique, permutation feature importance) aux plus complexes (celles basées sur la valeur de Shapley), toutes mettent en évidence l'importance des équipements de sécurité (notamment l'utilisation ou non de la ceinture de sécurité et le port ou non du casque), mais aussi le fait de rouler en agglomération ou non. On note aussi que la meilleure méthode, une forêt aléatoire, permet, comme les autres méthodes basées sur les arbres de classification, de prendre en compte l'effet non-linéaire des variables continues ainsi que les interactions entre variables. C'est probablement grâce à cette flexibilité qu'elle a pu surpasser la régression logistique : il apparaît en effet lors de l'interprétabilité du modèle que les variables continues « âge usager » et « longitude, latitude » ont une forte importance dans la forêt aléatoire, mais une importance faible dans la régression logistique, dans laquelle elles ont été introduites linéairement.

Nous pouvons effectivement vérifier dans la Figure 94 que la position géographique influe sur la probabilité qu'un accidenté soit hospitalisé ou tué, ceci de façon non-linéaire. Cette carte a été réalisée dans R avec un modèle bayésien à effet spatial aléatoire, et indique les zones où le risque d'accident grave excède le risque moyen national (rouge), est équivalent (jaune) ou est en deçà (vert). La plus faible gravité des accidents en agglomération est visible à l'œil nu, sur Paris notamment, alors que le modèle de lissage ne contient aucune information autre que la position géographique. Une piste d'amélioration du modèle pourrait donc être de modéliser plus précisément le processus spatial, soit avec une approche géostatistique pure, comme publié par (Barmoudeh, Baghislani, & Martino, 2022) pour classifier les accidents en Iran en 3 classes de gravité, soit en intégrant des composantes géospatiales à des modèles de machine learning de type « arbres de classification », comme proposé par (Effati & Sadeghi-Niaraki, 2015), en Iran aussi.

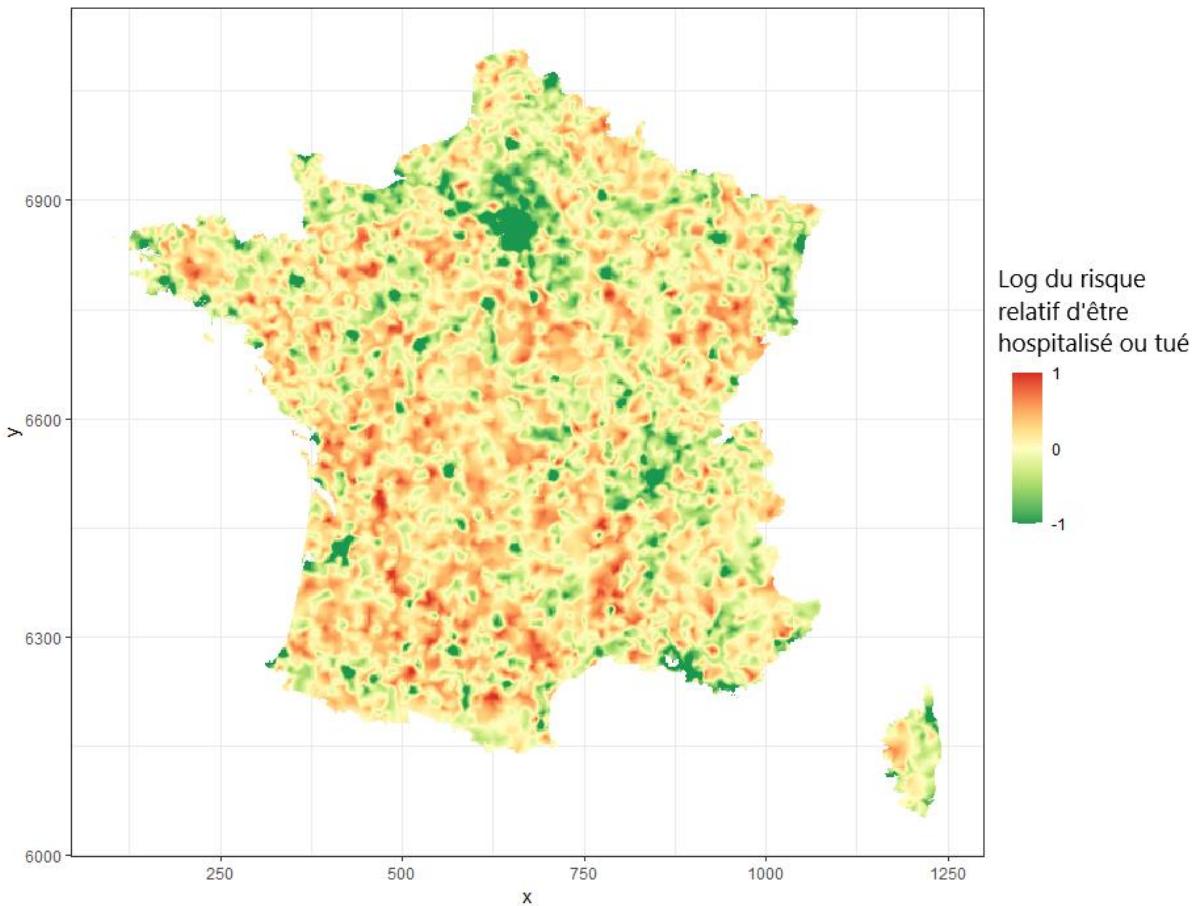


Figure 94. Risque relatif (échelle log) d'être hospitalisé ou tué pour une personne de la base de données

Finalement, nous avons pu observer une légère amélioration des résultats :

- augmentation de 2% des vrais positifs en séparant le jeu de données composé de 4 classes en 4 jeux de données binaires,
- augmentation de 0.14% des vrais positifs en modifiant les variables 'catv', 'obs', 'obsm' et en créant la variable 'catv_percute'.

Nous pouvons espérer qu'en combinant les 2 approches cela permettra encore une amélioration des résultats. En revanche, la seconde approche reste incomplète car nous ne disposons des types de véhicules qui ont percuté la personne accidentée que pour les piétons. C'est pourquoi nos résultats laissent penser que notre approche de modélisation pourrait être améliorée en incluant cette variable pour tous les accidentés. De plus, l'inclusion de variables telles que la vitesse au moment de l'accident, l'état de santé de la personne ou encore le fait d'être sous emprise de l'alcool ou de stupéfiants permettrait certainement d'affiner les prédictions.

V PREDICTION DE LA GRAVITE DE L'ACCIDENT – DEEP LEARNING

Lors de nos différentes modélisations, nous avons essayé 8 modèles de Machine Learning (régression logistique, SVM, Decision Tree, Random Forest, Balanced Random Forest, CatBoost, XGBoost et KNN) et nous avons réalisé des optimisations par ajout d'une variable et par classification binaire. Pour poursuivre la démarche scientifique, nous avons décidé d'essayer un modèle de Deep Learning, et de recourir, à des fins pédagogiques, aux deux grandes librairies usuellement utilisées dans ce cadre, à savoir Keras et Pytorch, ainsi qu'à une librairie, TabNet, développée pour appliquer des réseaux de neurones profonds sur des données tabulaires. Enfin, une nouvelle variable a été identifiée comme pouvant améliorer les performances des modèles présentés.

V.1 Modélisation par Deep Learning avec Keras

V.1.1 Modèle de référence

Nous appliquons les étapes de pre-processing décidées à l'issue de l'analyse préliminaire des données (section I.5) :

- o Les classes de la variable cible 'grav' sont renumérotées de 0 à 3 (0 - indemnes, 1 - Tués, 2 - Blessés hospitalisés, 3 - Blessés légers)
 - o Transformation des heures et des mois
 - o Transformation des latitudes et longitudes (RobustScaler)
 - o Transformation de l'âge (StandardScaler)

Le modèle choisi est le suivant :

Layer (type)	Output Shape	Param #
dense_6 (Dense)	(None, 70)	2,660
dense_7 (Dense)	(None, 140)	9,940
dropout_2 (Dropout)	(None, 140)	0
dense_8 (Dense)	(None, 70)	9,870
dense_9 (Dense)	(None, 35)	2,485
dropout_3 (Dropout)	(None, 35)	0
dense_10 (Dense)	(None, 14)	504
dense_11 (Dense)	(None, 4)	60

Chaque couche Dense est activée par la fonction d'activation ReLu sauf la dernière couche qui est activée par la fonction Softmax. Les couches de Dropout ont un rate de 0,2 afin de diminuer le surapprentissage.

Le modèle est compilé avec :

- loss : sparse_categorical_crossentropy
- optimizer: adam
- metrics : sparse_categorical_accuracy

Enfin le modèle est entraîné avec :

- epochs = 100
- batch_size = 512 (nous choisissons un nombre élevé afin que le modèle ait plus de chance de rencontrer la modalité des tués)
- validation_split = 0,1
- callbacks = [reduce_learning_rate]

Le `reduce_learning_rate` a préalablement été défini de la manière suivante :

```
reduce_learning_rate = ReduceLROnPlateau(monitor = 'val_loss',
                                         min_delta = 0.01,
                                         patience = 5,
                                         factor = 0.5,
                                         cooldown = 2,
                                         verbose = 1)
```

Ce modèle permet d'obtenir les résultats suivants :

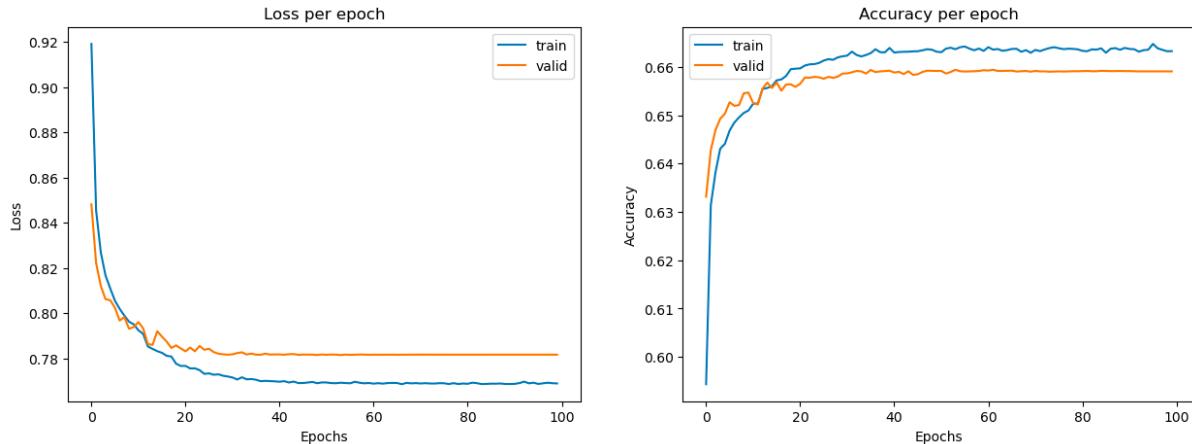


Figure 95 : Courbes de perte et d'accuracy en fonction du nombre d'époques pour le modèle de référence Keras

	precision	recall	f1-score	support		Classe prédictive	0	1	2	3
						Classe réelle				
0	0.73	0.80	0.76	46137		0	37000	14	1249	7874
1	0.46	0.05	0.10	3050		1	316	167	1663	904
2	0.50	0.39	0.44	17500		2	2047	144	6881	8428
3	0.63	0.65	0.64	45097		3	11561	35	4084	29417
accuracy			0.66	111784						
macro avg	0.58	0.48	0.49	111784						
weighted avg	0.64	0.66	0.64	111784						

Figure 96 : Métriques et matrice de confusion pour le modèle de référence (Keras)

On remarque que le déséquilibre du jeu de données impacte, comme pour les modèles de Machine Learning, aussi fortement les prédictions du modèle. En effet, les classes minoritaires (1,2) sont aussi les moins bien prédites et la classe 1 est très peu détectée. Nous décidons donc de regarder si un ré-équilibrage des classes permet d'améliorer les prédictions.

V.1.2 Rééquilibrage du jeu de données

Le jeu de données étant fortement déséquilibré, nous l'avons rééquilibré en utilisant 3 méthodes différentes :

- en ajoutant un `class_weight` pendant la phase d'entraînement du modèle,
- en diminuant le jeu de données des classes ayant le plus de lignes avec un `RandomUnderSampling`,
- en augmentant le jeu de données des classes ayant le moins de lignes avec un `RandomOverSampling`.

En réalisant la même compilation et entraînement que précédemment, nous obtenons les résultats présentés sur la Figure 97.

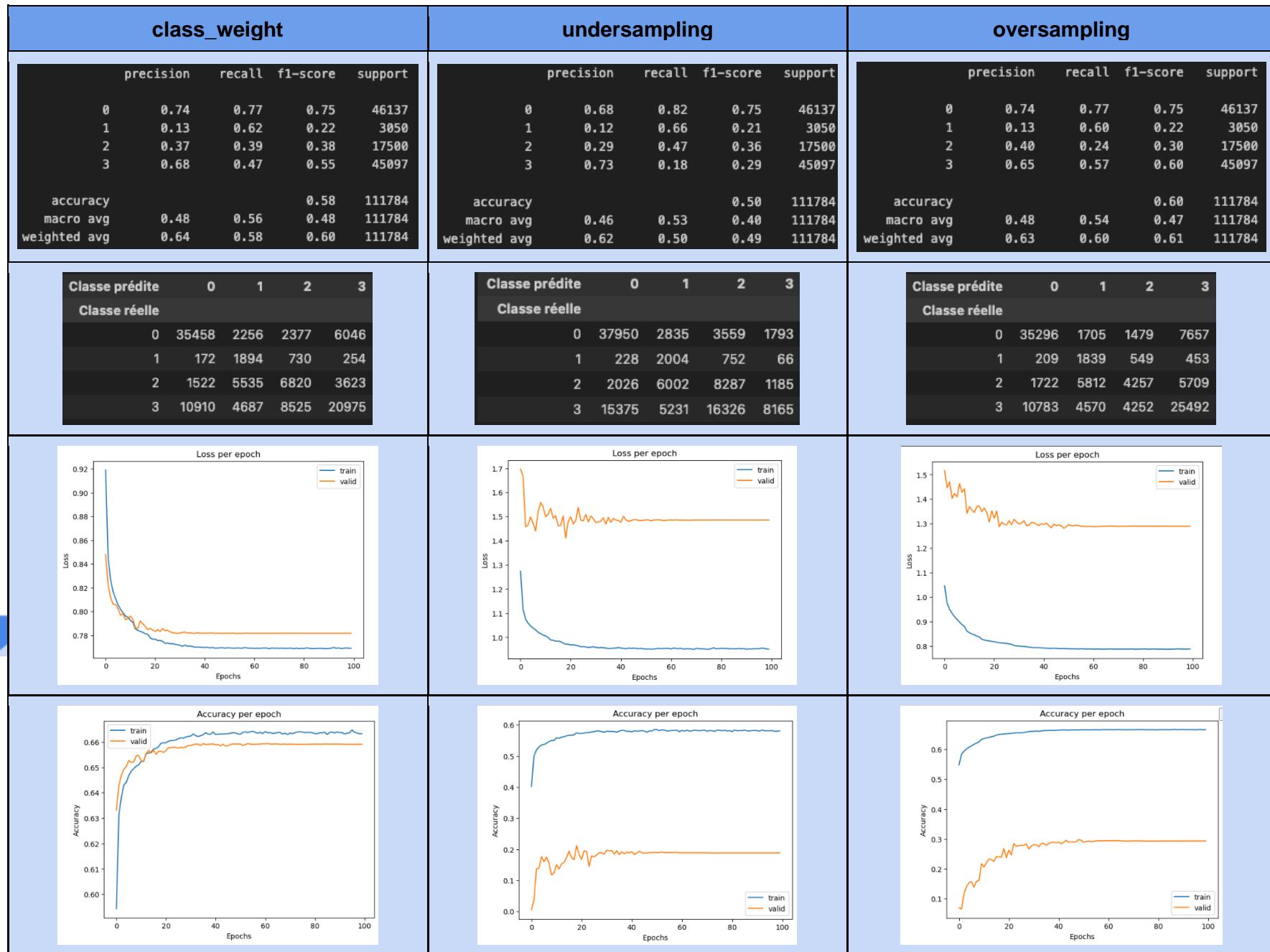


Figure 97 : Résultats des modèles Keras avec ré-équilibrage des classes

Le rééquilibrage du jeu de données permet d'augmenter le f1-score de la modalité tués (classe 1) dans tous les cas au détriment de l'accuracy. Cependant cette amélioration diminue aussi plus ou moins le f1-score des blessés hospitalisés et blessés légers selon le modèle.

Nous décidons donc d'optimiser les hyperparamètres afin d'améliorer encore les prédictions.

V.1.3 Hyperparamétrage du modèle

Pour essayer d'optimiser le modèle, nous décidons de rechercher les meilleurs paramètres pour :

- reduction de loss dans la compilation du modèle avec :

```
loss = tf.keras.losses.SparseCategoricalCrossentropy(  
    from_logits=False,  
    ignore_class=None,  
    reduction=reduction,  
    name="sparse_categorical_crossentropy",  
)
```
- activation des couches Denses (sauf pour la dernière couche qui est une activation Softmax)
- kernel_initializer ajouté ou non dans les couches Dense (sauf pour la dernière couche)
- batch_size dans l'entraînement du modèle
- epochs dans l'entraînement du modèle

Voici le tableau regroupant les différentes valeurs testées pour chacun de ces paramètres :

reduction de loss	sum, sum_over_batch_size, None
activation	relu, sigmoid, tanh, leaky_relu, swish, elu, selu, gelu
kernel_initializer	RandomNormal, RandomUniform, TruncatedNormal, GlorotNormal, GlorotUniform, HeNormal, HeUniform, Orthogonal, VarianceScaling_in, VarianceScaling_out, LecunNormal, LecunUniform
batch_size	32, 64, 128, 256, 512, 1024, 2048
epochs	10, 20, 30, 40, 50, 60, 70, 80, 90, 100

En réalisant ces hyperparamétrages nous obtenons les résultats présentés dans la figure ...

Les hyperparamétrages permettent d'augmenter l'accuracy de tous les modèles de 1% à 3%. Pour le modèle utilisant class_weight, ces modifications influencent peu les résultats. En revanche, cela permet d'augmenter notamment le f1-score des blessés légers dans le cas de l'undersampling et celui des blessés hospitalisés dans le cas de l'oversampling.

En comparant les modèles entre eux, on voit que l'undersampling obtient les moins bons résultats, alors que l'oversampling permet d'avoir la meilleure accuracy et les meilleurs f1-score. Cependant, les courbes de la loss par epoch et d'accuracy par epoch montrent un surapprentissage dans le cas de l'oversampling et de l'undersampling. On décide donc de conserver le modèle utilisant class_weight comme meilleur modèle car il n'y a pas de surapprentissage visible sur les courbes.

class_weight	undersampling	oversampling																																																																																																																								
reduction de loss = sum_over_batch_size activation = gelu, kernel_initializer = GlorotNormal batch_size = 128 epochs = 70	reduction de loss = sum activation = swish kernel_initializer = VarianceScaling_in batch_size = 64 epochs = 40	reduction de loss = sum_over_batch_size activation = swish, kernel_initializer = LecunNormal batch_size = 32 epochs = 70																																																																																																																								
<table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>0</td><td>0.75</td><td>0.76</td><td>0.75</td><td>46137</td></tr> <tr> <td>1</td><td>0.13</td><td>0.61</td><td>0.22</td><td>3050</td></tr> <tr> <td>2</td><td>0.37</td><td>0.40</td><td>0.38</td><td>17500</td></tr> <tr> <td>3</td><td>0.68</td><td>0.47</td><td>0.56</td><td>45097</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.59</td><td>111784</td></tr> <tr> <td>macro avg</td><td>0.48</td><td>0.56</td><td>0.48</td><td>111784</td></tr> <tr> <td>weighted avg</td><td>0.64</td><td>0.59</td><td>0.60</td><td>111784</td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.75	0.76	0.75	46137	1	0.13	0.61	0.22	3050	2	0.37	0.40	0.38	17500	3	0.68	0.47	0.56	45097	accuracy			0.59	111784	macro avg	0.48	0.56	0.48	111784	weighted avg	0.64	0.59	0.60	111784	<table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>0</td><td>0.70</td><td>0.82</td><td>0.75</td><td>46137</td></tr> <tr> <td>1</td><td>0.12</td><td>0.65</td><td>0.21</td><td>3050</td></tr> <tr> <td>2</td><td>0.31</td><td>0.47</td><td>0.38</td><td>17500</td></tr> <tr> <td>3</td><td>0.73</td><td>0.25</td><td>0.37</td><td>45097</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.53</td><td>111784</td></tr> <tr> <td>macro avg</td><td>0.47</td><td>0.55</td><td>0.43</td><td>111784</td></tr> <tr> <td>weighted avg</td><td>0.64</td><td>0.53</td><td>0.52</td><td>111784</td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.70	0.82	0.75	46137	1	0.12	0.65	0.21	3050	2	0.31	0.47	0.38	17500	3	0.73	0.25	0.37	45097	accuracy			0.53	111784	macro avg	0.47	0.55	0.43	111784	weighted avg	0.64	0.53	0.52	111784	<table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>0</td><td>0.73</td><td>0.78</td><td>0.75</td><td>46137</td></tr> <tr> <td>1</td><td>0.14</td><td>0.49</td><td>0.22</td><td>3050</td></tr> <tr> <td>2</td><td>0.41</td><td>0.37</td><td>0.39</td><td>17500</td></tr> <tr> <td>3</td><td>0.67</td><td>0.54</td><td>0.59</td><td>45097</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.61</td><td>111784</td></tr> <tr> <td>macro avg</td><td>0.49</td><td>0.54</td><td>0.49</td><td>111784</td></tr> <tr> <td>weighted avg</td><td>0.64</td><td>0.61</td><td>0.62</td><td>111784</td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.73	0.78	0.75	46137	1	0.14	0.49	0.22	3050	2	0.41	0.37	0.39	17500	3	0.67	0.54	0.59	45097	accuracy			0.61	111784	macro avg	0.49	0.54	0.49	111784	weighted avg	0.64	0.61	0.62	111784
	precision	recall	f1-score	support																																																																																																																						
0	0.75	0.76	0.75	46137																																																																																																																						
1	0.13	0.61	0.22	3050																																																																																																																						
2	0.37	0.40	0.38	17500																																																																																																																						
3	0.68	0.47	0.56	45097																																																																																																																						
accuracy			0.59	111784																																																																																																																						
macro avg	0.48	0.56	0.48	111784																																																																																																																						
weighted avg	0.64	0.59	0.60	111784																																																																																																																						
	precision	recall	f1-score	support																																																																																																																						
0	0.70	0.82	0.75	46137																																																																																																																						
1	0.12	0.65	0.21	3050																																																																																																																						
2	0.31	0.47	0.38	17500																																																																																																																						
3	0.73	0.25	0.37	45097																																																																																																																						
accuracy			0.53	111784																																																																																																																						
macro avg	0.47	0.55	0.43	111784																																																																																																																						
weighted avg	0.64	0.53	0.52	111784																																																																																																																						
	precision	recall	f1-score	support																																																																																																																						
0	0.73	0.78	0.75	46137																																																																																																																						
1	0.14	0.49	0.22	3050																																																																																																																						
2	0.41	0.37	0.39	17500																																																																																																																						
3	0.67	0.54	0.59	45097																																																																																																																						
accuracy			0.61	111784																																																																																																																						
macro avg	0.49	0.54	0.49	111784																																																																																																																						
weighted avg	0.64	0.61	0.62	111784																																																																																																																						
<table border="1"> <thead> <tr> <th>Classe prédictive</th><th>0</th><th>1</th><th>2</th><th>3</th></tr> <tr> <th>Classe réelle</th><th></th><th></th><th></th><th></th></tr> </thead> <tbody> <tr> <td>0</td><td>35171</td><td>2138</td><td>2595</td><td>6233</td></tr> <tr> <td>1</td><td>157</td><td>1870</td><td>782</td><td>241</td></tr> <tr> <td>2</td><td>1439</td><td>5422</td><td>7054</td><td>3585</td></tr> <tr> <td>3</td><td>10417</td><td>4597</td><td>8776</td><td>21307</td></tr> </tbody> </table>	Classe prédictive	0	1	2	3	Classe réelle					0	35171	2138	2595	6233	1	157	1870	782	241	2	1439	5422	7054	3585	3	10417	4597	8776	21307	<table border="1"> <thead> <tr> <th>Classe prédictive</th><th>0</th><th>1</th><th>2</th><th>3</th></tr> <tr> <th>Classe réelle</th><th></th><th></th><th></th><th></th></tr> </thead> <tbody> <tr> <td>0</td><td>37662</td><td>2684</td><td>3443</td><td>2348</td></tr> <tr> <td>1</td><td>196</td><td>1972</td><td>788</td><td>94</td></tr> <tr> <td>2</td><td>1867</td><td>5741</td><td>8307</td><td>1585</td></tr> <tr> <td>3</td><td>14415</td><td>5448</td><td>14103</td><td>11131</td></tr> </tbody> </table>	Classe prédictive	0	1	2	3	Classe réelle					0	37662	2684	3443	2348	1	196	1972	788	94	2	1867	5741	8307	1585	3	14415	5448	14103	11131	<table border="1"> <thead> <tr> <th>Classe prédictive</th><th>0</th><th>1</th><th>2</th><th>3</th></tr> <tr> <th>Classe réelle</th><th></th><th></th><th></th><th></th></tr> </thead> <tbody> <tr> <td>0</td><td>35768</td><td>1406</td><td>2179</td><td>6784</td></tr> <tr> <td>1</td><td>223</td><td>1501</td><td>924</td><td>402</td></tr> <tr> <td>2</td><td>1723</td><td>4298</td><td>6474</td><td>5005</td></tr> <tr> <td>3</td><td>11261</td><td>3251</td><td>6359</td><td>24226</td></tr> </tbody> </table>	Classe prédictive	0	1	2	3	Classe réelle					0	35768	1406	2179	6784	1	223	1501	924	402	2	1723	4298	6474	5005	3	11261	3251	6359	24226																														
Classe prédictive	0	1	2	3																																																																																																																						
Classe réelle																																																																																																																										
0	35171	2138	2595	6233																																																																																																																						
1	157	1870	782	241																																																																																																																						
2	1439	5422	7054	3585																																																																																																																						
3	10417	4597	8776	21307																																																																																																																						
Classe prédictive	0	1	2	3																																																																																																																						
Classe réelle																																																																																																																										
0	37662	2684	3443	2348																																																																																																																						
1	196	1972	788	94																																																																																																																						
2	1867	5741	8307	1585																																																																																																																						
3	14415	5448	14103	11131																																																																																																																						
Classe prédictive	0	1	2	3																																																																																																																						
Classe réelle																																																																																																																										
0	35768	1406	2179	6784																																																																																																																						
1	223	1501	924	402																																																																																																																						
2	1723	4298	6474	5005																																																																																																																						
3	11261	3251	6359	24226																																																																																																																						

Figure 98 : Résultats des modèles de Deep Learning (Keras) optimisés

V.2 Modélisation par Deep Learning avec PyTorch

Ce projet étant réalisé en parallèle de la formation, il nous a semblé intéressant de profiter de ce projet pour prendre également en main la librairie PyTorch, moins développée dans les supports de cours qui nous sont proposés.

V.2.1 Modèle de référence

Les étapes de preprocessing sont les suivantes :

- o Les classes de la variable cible ‘grav’ sont re-numérotées de 0 à 3, et **ré-arrangées dans un ordre croissant de gravité** (0 - indemnes, 1 - Blessés légers, 2 - Blessés hospitalisés, 3 - Tués)
- o Transformation des heures et des mois
- o Transformation des latitudes et longitudes (RobustScaler)
- o Transformation de l’âge (MinMaxScaler)

La succession des couches du modèle de référence est la même que celle utilisée avec Keras, et de même, chaque couche dense est filtrée en sortie de couche par une fonction ReLU, sauf la dernière qui a une fonction d’activation de type softmax de façon à obtenir les probabilités d’appartenance aux différentes classes.

Sont utilisés une fonction de perte d’entropie croisée de classification (CrossEntropyLoss), avec une réduction de cette fonction sur les mini-lots d’échantillons de type ‘mean’, et un optimiseur SGD. Dans l’optimiseur, le taux d’apprentissage, initialement de 0.1, fait l’objet d’un rappel de type ReduceLROnPlateau qui permet de réduire ce taux d’apprentissage après un certain nombre d’époques. Ici, la patience est de 2. Après 2 époques, sans amélioration du f1-score, la réduction du taux est d’un facteur 0,25 (mais toujours supérieur à 1e-4).

La taille des batchs est de 64.

La Figure 99 présente l’évolution de la précision, et de la fonction de perte en fonction du nombre d’époques pour les échantillons d’entraînement et de validation.

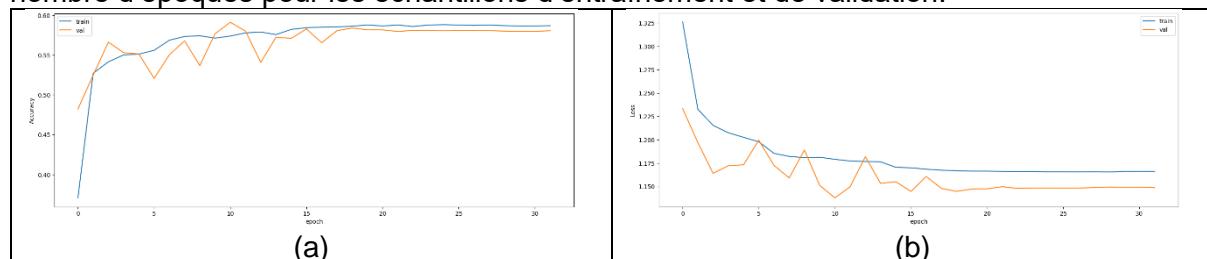


Figure 99 : Courbes de perte et d’accuracy selon le nombre d’époques pour le modèle de référence (PyTorch)

La Figure 100 présente les métriques des différentes classes, ainsi que la matrice de confusion du modèle. Ces résultats sont similaires à ceux obtenus avec Keras, ce qui nous conforte dans nos utilisations des deux librairies. Le constat est donc le même que précédemment, l’utilisation de deep learning sur nos données tabulaires ne conduit pas à des performances plus intéressantes que le machine learning.

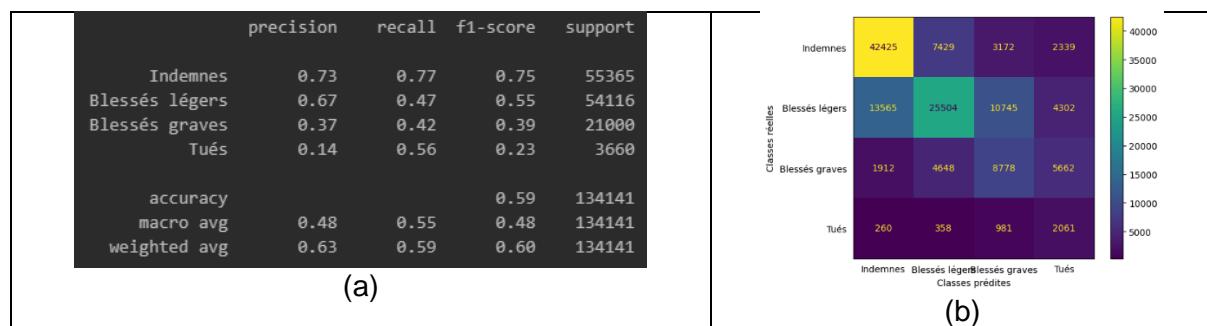


Figure 100 : Métriques et matrice de confusion pour le modèle Deep Learning de référence (PyTorch)

V.2.2 Hyperparamétrage

L'analyse des hyperparamètres a été conduite avec la librairie **Skorch**, qui rend possible l'utilisation de PyTorch avec scikit-learn (et donc GridSearchCV).

Pour chaque hyperparamètre, la comparaison des performances est réalisée par validation croisée sur 3 échantillons, issus de l'ensemble d'entraînement.

Recherche de la meilleure fonction d'activation

Le Tableau 18 : Comparaison des performances selon la fonction d'activation choisie fournit les moyennes et écarts-types des performances obtenues sur les 3 échantillons testés pour chaque fonction d'activation testée. Les meilleurs résultats sont obtenus avec la **fonction ReLU**, suivie de près par la fonction tanh qui conduit cependant à un écart-type plus important.

Tableau 18 : Comparaison des performances selon la fonction d'activation choisie

Fonction d'activation	ReLU	GELU	Softplus	Softsign	Tanh
Accuracy	0.5041 ± 0.0183	0.4514 ± 0.0129	0.4102 ± 0.0047	0.5003 ± 0.0266	0.5035 ± 0.0295

Recherche du meilleur taux de dropout et du meilleur nombre de neurones

La Figure 101 (a) fournit les performances pour chaque taux de dropout testé. Avec les barres de représentation des écarts-types, on s'aperçoit que le taux de dropout a relativement peu d'influence sur la performance du modèle. Un léger dropout semble cependant permettre de diminuer l'écart type en fonction de l'échantillon considéré.

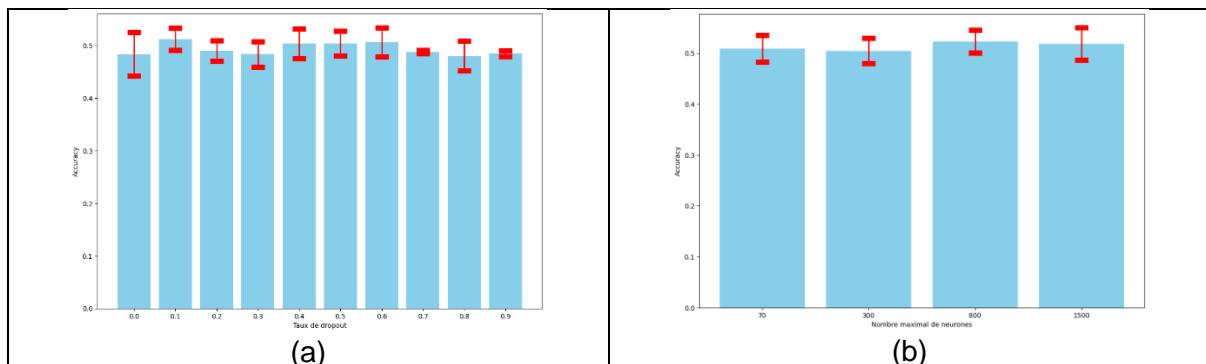


Figure 101 : Comparaison des performances en fonction (a) du taux de dropout choisi en sortie de couche Dense, (b) du nombre maximal de neurones dans le modèle

La Figure 101 (b) analyse l'influence du nombre maximal de neurones sur les performances. La deuxième couche dense du modèle de référence est celle possédant le plus grand nombre de neurones. Nous avons choisi de modifier ce paramètre, en conservant ensuite l'architecture globale du modèle (50% de neurones dans la couche 3, etc). Sur une plage de variations de 70 à 1500 neurones, il apparaît que ce paramètre impacte peu la performance globale du modèle.

V.3 Modélisation par Deep Learning avec TabNet

TabNet est une architecture d'apprentissage de données tabulaires profondes qui utilise un mécanisme d'attention séquentielle pour choisir les caractéristiques à chaque étape de décision.

Nous appliquons les étapes de preprocessing décidées à l'issue de l'analyse préliminaire des données :

- o Les classes de la variable cible ‘grav’ sont re-numérotées de 0 à 3, et **ré-arrangées dans un ordre croissant de gravité** (0 - indemnes, 1 - Blessés légers, 2 - Blessés hospitalisés, 3 - Tués)
- o Transformation des heures et des mois
- o Transformation des latitudes et longitudes (RobustScaler)
- o Transformation de l’âge (MinMaxScaler)

V.3.1 Optimisation des hyperparamètres avec Optuna

Le graphique à coordonnées parallèles de la Figure 102 présente les hyperparamètres analysés ainsi que les performances obtenues. L’analyse des importances des hyperparamètres montre que le nombre maximal d’époques et le gamma sont les 2 paramètres les plus influents parmi ceux analysés. Il est préférable de choisir un nombre maximal d’époques supérieur à 30 et une valeur de gamma relativement faible.

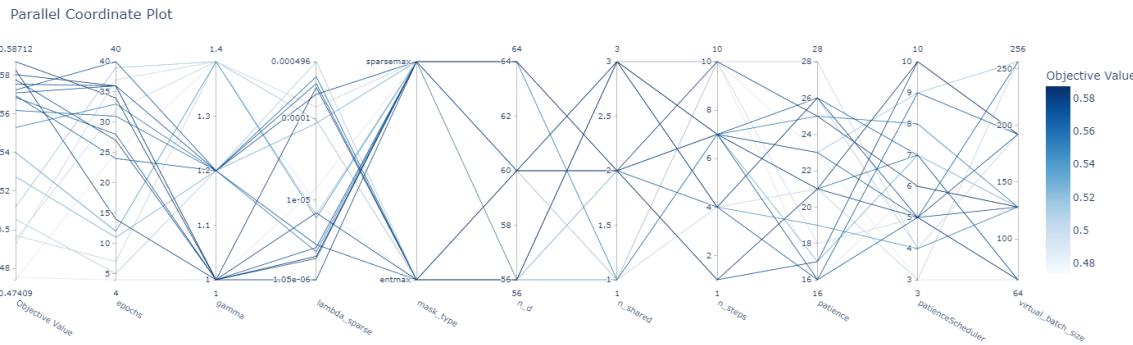


Figure 102 : Parallel Coordinate Plot pour l’hyperparamétrage du modèle TabNet avec Optuna

Les meilleurs paramètres, parmi les 20 configurations testées, sont ceux indiqués dans le Tableau 19. Le lecteur est invité à se rendre sur la documentation de TabNet (<https://dreamquark-ai.github.io/tabnet/>) pour la signification de ces différents paramètres.

Tableau 19 : Paramètres du modèle TabNet après optimisation

mask_type	n_d	n_steps	gamma	n_shared	lambda_sparse	patience (scheduler)	max_epochs	patience	virtual_batch_size
sparsemax	64	1	1.0	2	2.0399e-06	10	34	21	192

Les performances du modèle avec ces paramètres sont données à la Figure 103. On retrouve ici un modèle de Deep Learning, relativement performant, en comparaison de certains modèles de Machine Learning, mais qui cependant n’atteint pas les performances obtenues avec les modèles de type forêt aléatoire ou catboost.

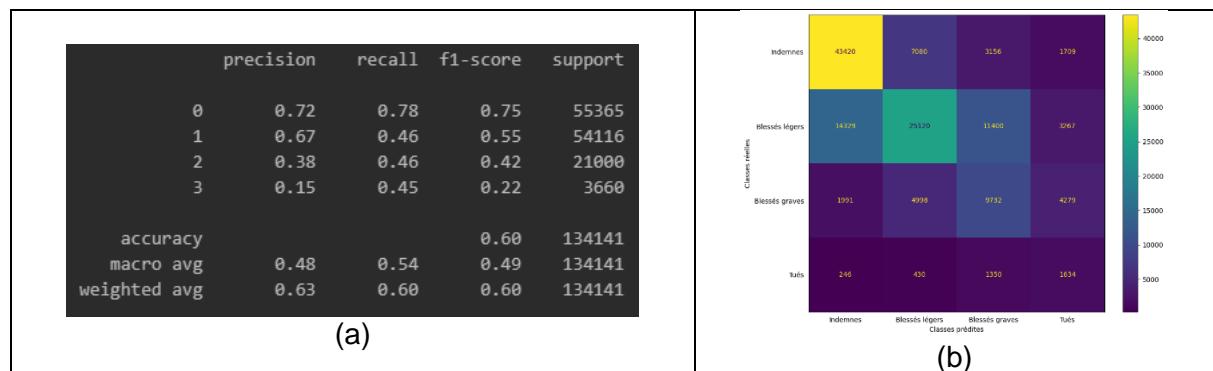


Figure 103 : Métriques et matrice de confusion pour le modèle TabNet optimisé

V.3.2 Interprétabilité du modèle

Une feature importance est intégrée au modèle TabNet. La Figure 104 (a) indique que la catégorie de route, la présence d'un obstacle mobile, les équipements de sécurité (gilet et airbag) et la place occupée par l'usager sont les variables sur lesquelles le modèle a porté son attention le plus fréquemment. La Figure 104 (b) montre le masque d'activation des variables (un seul, puisque $n_steps = 1$) pour les 50 premières observations de l'ensemble de test.

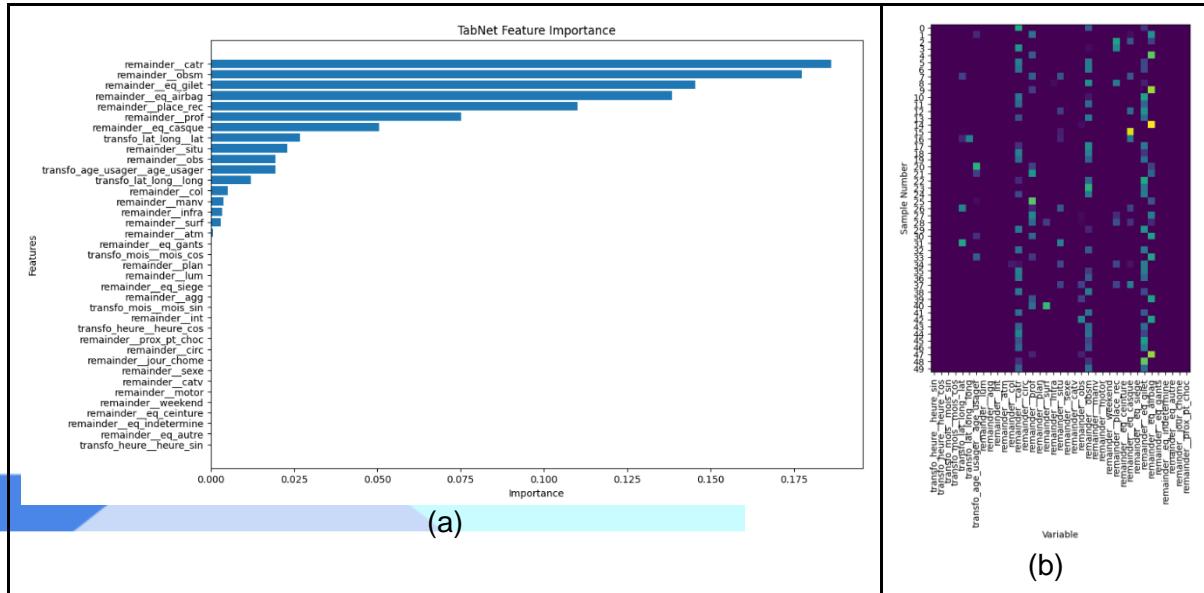


Figure 104 : (a) Importance des variables dans le modèle TabNet (b). Masque d'activation des variables pour les 50 premières observations de l'échantillon de test.

V.4 Comparaison Deep Learning / Machine Learning (classification multi-classes)

Les modèles de Deep Learning que nous avons mis en place obtiennent de moins bons résultats que le meilleur modèle de Machine Learning.

Meilleur modèle de Machine Learning (Random Forest Classifier)					Meilleur modèle de Deep Learning (class_weight)																																																																																				
<table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>1</td><td>0.70</td><td>0.85</td><td>0.77</td><td>46137</td></tr> <tr> <td>2</td><td>0.17</td><td>0.49</td><td>0.26</td><td>3050</td></tr> <tr> <td>3</td><td>0.41</td><td>0.49</td><td>0.45</td><td>17500</td></tr> <tr> <td>4</td><td>0.74</td><td>0.42</td><td>0.54</td><td>45097</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.61</td><td>111784</td></tr> <tr> <td>macro avg</td><td>0.51</td><td>0.56</td><td>0.50</td><td>111784</td></tr> <tr> <td>weighted avg</td><td>0.66</td><td>0.61</td><td>0.61</td><td>111784</td></tr> </tbody> </table>						precision	recall	f1-score	support	1	0.70	0.85	0.77	46137	2	0.17	0.49	0.26	3050	3	0.41	0.49	0.45	17500	4	0.74	0.42	0.54	45097	accuracy			0.61	111784	macro avg	0.51	0.56	0.50	111784	weighted avg	0.66	0.61	0.61	111784	<table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>Indemnés</td><td>0.72</td><td>0.78</td><td>0.75</td><td>55365</td></tr> <tr> <td>Blessés légers</td><td>0.67</td><td>0.46</td><td>0.55</td><td>54116</td></tr> <tr> <td>Blessés graves</td><td>0.38</td><td>0.46</td><td>0.42</td><td>21000</td></tr> <tr> <td>Tués</td><td>0.15</td><td>0.45</td><td>0.22</td><td>3660</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.60</td><td>134141</td></tr> <tr> <td>macro avg</td><td>0.48</td><td>0.54</td><td>0.49</td><td>134141</td></tr> <tr> <td>weighted avg</td><td>0.63</td><td>0.60</td><td>0.60</td><td>134141</td></tr> </tbody> </table>						precision	recall	f1-score	support	Indemnés	0.72	0.78	0.75	55365	Blessés légers	0.67	0.46	0.55	54116	Blessés graves	0.38	0.46	0.42	21000	Tués	0.15	0.45	0.22	3660	accuracy			0.60	134141	macro avg	0.48	0.54	0.49	134141	weighted avg	0.63	0.60	0.60	134141
	precision	recall	f1-score	support																																																																																					
1	0.70	0.85	0.77	46137																																																																																					
2	0.17	0.49	0.26	3050																																																																																					
3	0.41	0.49	0.45	17500																																																																																					
4	0.74	0.42	0.54	45097																																																																																					
accuracy			0.61	111784																																																																																					
macro avg	0.51	0.56	0.50	111784																																																																																					
weighted avg	0.66	0.61	0.61	111784																																																																																					
	precision	recall	f1-score	support																																																																																					
Indemnés	0.72	0.78	0.75	55365																																																																																					
Blessés légers	0.67	0.46	0.55	54116																																																																																					
Blessés graves	0.38	0.46	0.42	21000																																																																																					
Tués	0.15	0.45	0.22	3660																																																																																					
accuracy			0.60	134141																																																																																					
macro avg	0.48	0.54	0.49	134141																																																																																					
weighted avg	0.63	0.60	0.60	134141																																																																																					
Support : 25% de l'ensemble des données					Support : 30% de l'ensemble des données																																																																																				
<table border="1"> <thead> <tr> <th>Classes prédites</th><th>1</th><th>2</th><th>3</th><th>4</th></tr> <tr> <th>Classes réelles</th><th></th><th></th><th></th><th></th></tr> </thead> <tbody> <tr> <td>1</td><td>39232</td><td>1262</td><td>2339</td><td>3304</td></tr> <tr> <td>2</td><td>288</td><td>1488</td><td>1016</td><td>258</td></tr> <tr> <td>3</td><td>2225</td><td>3578</td><td>8632</td><td>3065</td></tr> <tr> <td>4</td><td>14677</td><td>2275</td><td>9058</td><td>19087</td></tr> </tbody> </table>					Classes prédites	1	2	3	4	Classes réelles					1	39232	1262	2339	3304	2	288	1488	1016	258	3	2225	3578	8632	3065	4	14677	2275	9058	19087	<table border="1"> <thead> <tr> <th>Classe prédite</th><th>0</th><th>1</th><th>2</th><th>3</th></tr> <tr> <th>Classe réelle</th><th></th><th></th><th></th><th></th></tr> </thead> <tbody> <tr> <td>0</td><td>43420</td><td>7080</td><td>3156</td><td>1709</td></tr> <tr> <td>1</td><td>14329</td><td>25120</td><td>11400</td><td>3267</td></tr> <tr> <td>2</td><td>1991</td><td>4998</td><td>9732</td><td>4279</td></tr> <tr> <td>3</td><td>246</td><td>430</td><td>1350</td><td>1634</td></tr> </tbody> </table>					Classe prédite	0	1	2	3	Classe réelle					0	43420	7080	3156	1709	1	14329	25120	11400	3267	2	1991	4998	9732	4279	3	246	430	1350	1634																				
Classes prédites	1	2	3	4																																																																																					
Classes réelles																																																																																									
1	39232	1262	2339	3304																																																																																					
2	288	1488	1016	258																																																																																					
3	2225	3578	8632	3065																																																																																					
4	14677	2275	9058	19087																																																																																					
Classe prédite	0	1	2	3																																																																																					
Classe réelle																																																																																									
0	43420	7080	3156	1709																																																																																					
1	14329	25120	11400	3267																																																																																					
2	1991	4998	9732	4279																																																																																					
3	246	430	1350	1634																																																																																					
1 - indemnés, 2 - Tués, 3 - Blessés hospitalisés, 4 - Blessés légers					0 - indemnés, 1 - Blessés légers, 2 - Blessés hospitalisés, 3 - Tués																																																																																				

Le fait que les modèles de Deep Learning ne rivalisent pas forcément avec les modèles de Machine Learning sur des données tabulaires a fait l'objet de nombreuses publications.

VI RECENTS ESSAIS D'OPTIMISATION – AJOUT D'UNE NOUVELLE VARIABLE

Nous avons déjà fait des essais d'optimisation des prédictions en réalisant des modélisations avec classifications binaires ou par ajout de la variable 'catv_percute'.

VI.1 Création de la variable 'nb_usagers_gr'

Nous décidons d'essayer d'ajouter une variable nombre d'usagers ('nb_usagers') en calculant, avant même de supprimer des lignes de notre jeu de données, le nombre de personnes accidentées lors d'un accident. Ceci peut facilement être connu grâce à la variable 'Num_Acc' (numéro de l'accident) qui est reprise pour chaque personne impliquée dans l'accident. La visualisation de cette variable permet de voir que la majorité des accidents impliquent 2 personnes (Figure 105 (a)) et que l'on obtient 90% des personnes accidentées en prenant les accidents impliquant de 1 à 5 personnes (Figure 105 (b)).

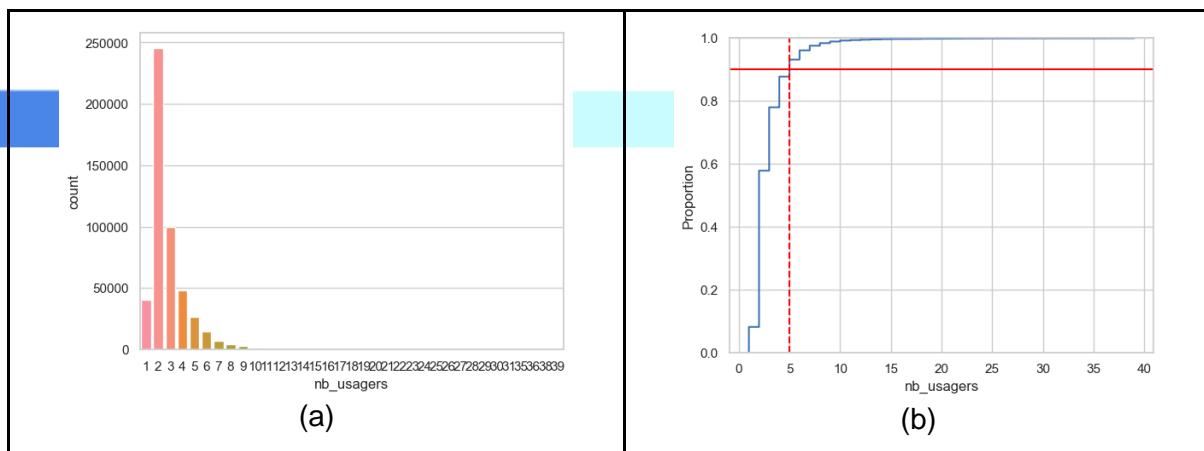


Figure 105 : Histogramme du nombre d'usagers impliqués dans les accidents, et fonction de répartition de cette variable.

De ce fait, nous décidons de créer une nouvelle variable 'nb_usagers_gr' qui conserve les modalités suivantes :

- 1 ⇒ 1 usager
- 2 ⇒ 2 usagers
- 3 ⇒ 3 usagers
- 4 ⇒ 4 usagers
- 5 ⇒ 5 ou + usagers

VI.2 Modélisation

Nous reprenons le meilleur modèle que nous avons obtenu pour la classification multiconcours : un algorithme de Random forest.

VI.3 Résultats et comparaison avec le jeu de données initial

La Figure 106 permet la comparaison des résultats des modèles Random Forest avec, et sans prise en compte de cette nouvelle variable.

Jeu de données initial	Jeu de données avec ajout de 'nb_usagers_gr'																																																																																
Random Forest (Train accuracy = 65.52%) (Test accuracy = 61.22%)	Random Forest (Train accuracy = 65,99%) (Test accuracy = 61,67%)																																																																																
paramètres : {'bootstrap': True, 'class_weight': 'balanced', 'criterion': 'entropy', 'max_depth': 13, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100, 'n_jobs': -1, 'random_state': 42}	paramètres : {'bootstrap': True, 'class_weight': 'balanced', 'criterion': 'entropy', 'max_depth': 13, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100, 'n_jobs': -1, 'random_state': 42}																																																																																
<table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>1</td><td>0.70</td><td>0.85</td><td>0.77</td><td>46137</td></tr> <tr> <td>2</td><td>0.17</td><td>0.49</td><td>0.26</td><td>3050</td></tr> <tr> <td>3</td><td>0.41</td><td>0.49</td><td>0.45</td><td>17500</td></tr> <tr> <td>4</td><td>0.74</td><td>0.42</td><td>0.54</td><td>45097</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.61</td><td>111784</td></tr> <tr> <td>macro avg</td><td>0.51</td><td>0.56</td><td>0.50</td><td>111784</td></tr> <tr> <td>weighted avg</td><td>0.66</td><td>0.61</td><td>0.61</td><td>111784</td></tr> </tbody> </table>		precision	recall	f1-score	support	1	0.70	0.85	0.77	46137	2	0.17	0.49	0.26	3050	3	0.41	0.49	0.45	17500	4	0.74	0.42	0.54	45097	accuracy			0.61	111784	macro avg	0.51	0.56	0.50	111784	weighted avg	0.66	0.61	0.61	111784	<table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>1</td><td>0.69</td><td>0.86</td><td>0.77</td><td>46137</td></tr> <tr> <td>2</td><td>0.18</td><td>0.48</td><td>0.27</td><td>3050</td></tr> <tr> <td>3</td><td>0.41</td><td>0.50</td><td>0.45</td><td>17500</td></tr> <tr> <td>4</td><td>0.75</td><td>0.42</td><td>0.54</td><td>45097</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.62</td><td>111784</td></tr> <tr> <td>macro avg</td><td>0.51</td><td>0.57</td><td>0.51</td><td>111784</td></tr> <tr> <td>weighted avg</td><td>0.66</td><td>0.62</td><td>0.61</td><td>111784</td></tr> </tbody> </table>		precision	recall	f1-score	support	1	0.69	0.86	0.77	46137	2	0.18	0.48	0.27	3050	3	0.41	0.50	0.45	17500	4	0.75	0.42	0.54	45097	accuracy			0.62	111784	macro avg	0.51	0.57	0.51	111784	weighted avg	0.66	0.62	0.61	111784
	precision	recall	f1-score	support																																																																													
1	0.70	0.85	0.77	46137																																																																													
2	0.17	0.49	0.26	3050																																																																													
3	0.41	0.49	0.45	17500																																																																													
4	0.74	0.42	0.54	45097																																																																													
accuracy			0.61	111784																																																																													
macro avg	0.51	0.56	0.50	111784																																																																													
weighted avg	0.66	0.61	0.61	111784																																																																													
	precision	recall	f1-score	support																																																																													
1	0.69	0.86	0.77	46137																																																																													
2	0.18	0.48	0.27	3050																																																																													
3	0.41	0.50	0.45	17500																																																																													
4	0.75	0.42	0.54	45097																																																																													
accuracy			0.62	111784																																																																													
macro avg	0.51	0.57	0.51	111784																																																																													
weighted avg	0.66	0.62	0.61	111784																																																																													
<table border="1"> <thead> <tr> <th>Classes prédites</th><th>1</th><th>2</th><th>3</th><th>4</th></tr> <tr> <th>Classes réelles</th><th></th><th></th><th></th><th></th></tr> </thead> <tbody> <tr> <td>1</td><td>39232</td><td>1262</td><td>2339</td><td>3304</td></tr> <tr> <td>2</td><td>288</td><td>1488</td><td>1016</td><td>258</td></tr> <tr> <td>3</td><td>2225</td><td>3578</td><td>8632</td><td>3065</td></tr> <tr> <td>4</td><td>14677</td><td>2275</td><td>9058</td><td>19087</td></tr> </tbody> </table>	Classes prédites	1	2	3	4	Classes réelles					1	39232	1262	2339	3304	2	288	1488	1016	258	3	2225	3578	8632	3065	4	14677	2275	9058	19087	<table border="1"> <thead> <tr> <th>Classes prédites</th><th>1</th><th>2</th><th>3</th><th>4</th></tr> <tr> <th>Classes réelles</th><th></th><th></th><th></th><th></th></tr> </thead> <tbody> <tr> <td>1</td><td>39779</td><td>1115</td><td>2337</td><td>2906</td></tr> <tr> <td>2</td><td>279</td><td>1462</td><td>1040</td><td>269</td></tr> <tr> <td>3</td><td>2249</td><td>3373</td><td>8732</td><td>3146</td></tr> <tr> <td>4</td><td>15016</td><td>2029</td><td>9088</td><td>18964</td></tr> </tbody> </table>	Classes prédites	1	2	3	4	Classes réelles					1	39779	1115	2337	2906	2	279	1462	1040	269	3	2249	3373	8732	3146	4	15016	2029	9088	18964																				
Classes prédites	1	2	3	4																																																																													
Classes réelles																																																																																	
1	39232	1262	2339	3304																																																																													
2	288	1488	1016	258																																																																													
3	2225	3578	8632	3065																																																																													
4	14677	2275	9058	19087																																																																													
Classes prédites	1	2	3	4																																																																													
Classes réelles																																																																																	
1	39779	1115	2337	2906																																																																													
2	279	1462	1040	269																																																																													
3	2249	3373	8732	3146																																																																													
4	15016	2029	9088	18964																																																																													
61.22% de vrais positifs	61,67% de vrais positifs																																																																																

Figure 106 : Comparaison des résultats des modèles Random Forest, avec ou sans la variable nb_usagers_gr

En ajoutant cette variable, on note une augmentation de l'accuracy de 0,45% qui se traduit par une légère augmentation du f1-score de la modalité tués. Ainsi il peut donc être intéressant d'ajouter la variable 'nb_usagers_gr' en plus des autres optimisations déjà évoquées (ajout de la variable 'catv_percute' et classification binaire).

VII CONCLUSIONS ET PERSPECTIVES

VII.1 Objectifs atteints

Nos deux objectifs étaient de prédire le nombre d'accidentés dans chaque classe de gravité selon la date et de proposer un modèle permettant de classer un accidenté dans l'une des 4 classes de gravité. Ils ont tous les deux été atteints.

Pour rappel, les différents modèles développés dans ce projet ont conduit aux performances de la Figure 107, jugées honorables.

Pour ces différents modèles, nous avons également identifié les variables explicatives les plus importantes pour déterminer l'état de gravité d'une personne accidentée (Figure 108). Ces résultats peuvent notamment être utilisés à des fins de prévention, et de sensibilisation des usagers de la route puisqu'ils mettent en évidence les circonstances qui orientent la gravité de l'accident. Nous avons ainsi démontré, entre autres, l'importance du port de la ceinture de sécurité et de la circulation en agglomération pour réduire la gravité des accidents.

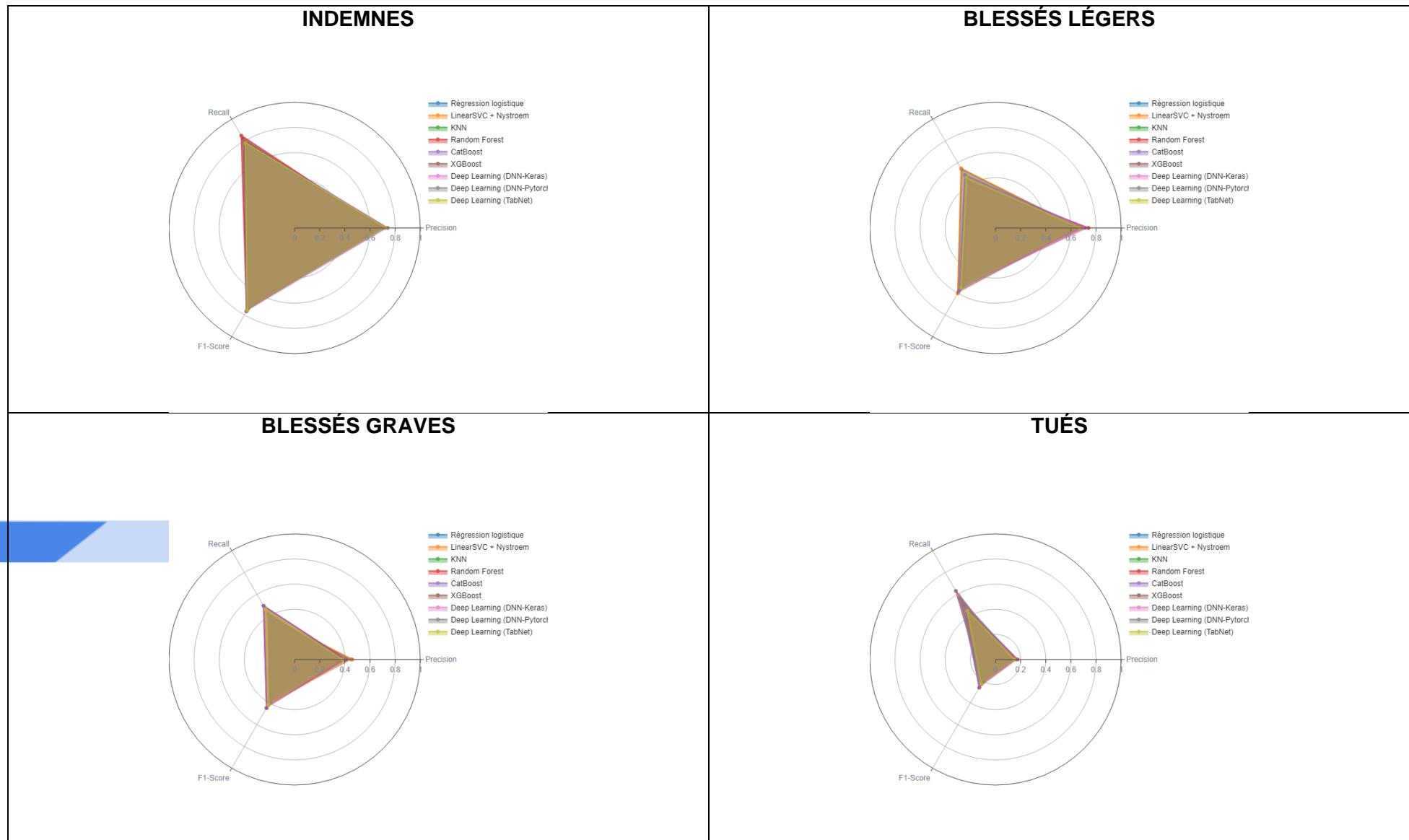


Figure 107 : Comparaison des performances des modèles de Machine Learning et de Deep Learning, pour chaque état de gravité

	ORDRE D'IMPORTANCE DES VARIABLES				
	Régression Logistique*	Forêts aléatoires	XGBoost (weight)	CatBoost	TabNet
Utilisation d'une ceinture de sécurité		1	1	1	
Age de l'usager		2	3	9	11
Type de collision	x	3		2	13
Accident survenu en agglomération		4	2	10	
Latitude		5	4	4	8
Place de l'usager dans véhicule (ou piéton)	x	6		5	5
Présence d'un obstacle mobile	xx	7	9	6	2
Présence d'un obstacle fixe	x	8	5	13	10
Catégorie de véhicule	xxxx	9		3	
Longitude		10		8	12
Utilisation d'un casque		11		27	7
Catégorie de route		12		7	1
Situation de l'accident	x	13		18	9
Heure à laquelle l'accident s'est produit		14		12	
Régime de circulation		15	6	19	
Proximité par rapport au point de choc		16		17	
Mois auquel est survenu l'accident		17		16	19
Sexe de l'usager	x	18	7	15	
Lumière		19		14	21
Motorisation du véhicule	x	20		24	
Manœuvre principale avant l'accident		21		11	14
Impossibilité de déterminer l'utilisation d'équipements de sécurité		22	8	21	
Présence d'une intersection		23		22	
Utilisation de gants		24		31	18
Infrastructure		25		28	15
Tracé en plan de la route (rectiligne ou non)		26		25	20
Etat de la surface de chaussée		27		23	16
Accident survenu un jour de week-end		28		20	
Conditions atmosphériques (normales/dégradées)		29		30	17
Profil en long de la route		30		26	6
Accident survenu un jour chômé		31		29	
Utilisation d'un équipement de sécurité (autre)		32		34	
Utilisation d'un airbag	x	33		32	4
Utilisation d'un gilet de sécurité		34		35	3
Utilisation d'un siège	x			33	

* La régression logistique s'appuyant sur les variables encodées, une croix est mise dès que l'une des modalités apparaît dans les variables associées aux plus forts coefficients

Figure 108 : Ordre d'importance des variables explicatives dans les modèles

VII.2 Difficultés rencontrées lors du projet

Lors de ce projet, nous avons principalement été confrontés à 4 problématiques :

- La gestion du déséquilibre des classes : les modalités tués et blessés hospitalisés sont très largement minoritaires dans le jeu de données. Pour pallier cela, nous avons essayé différentes approches de rééquilibrage des modalités (undersampling, oversampling, class_weight).
- La classification multiclass : la variable cible possède 4 modalités. Nous avons décidé de réaliser des prédictions en utilisant une classification multiclass. Cependant, lors des essais d'optimisation, nous avons pu constater que la séparation de notre jeu de données en 4 jeux de données binaires permettait d'obtenir de meilleurs résultats.
- La multiplicité des modalités pour les variables catégorielles : le jeu de données comporte majoritairement des variables catégorielles dont certaines peuvent contenir jusqu'à 40 modalités. Cette problématique a été traitée en réalisant des regroupements de modalité.
- L'interprétation personnelle de certaines variables : certaines variables, comme la variable 'atm' qui indique les conditions atmosphériques ou 'surf' qui indique l'état de surface de la route, restent à l'appréciation des forces de l'ordre lors du recueil des données.

De plus, nous avons rencontré différentes difficultés :

- Prévisionnelles :
 - Les interprétations par SHAP sur les random forest ont nécessité un temps beaucoup plus long que prévu,
 - Le timing du rendu des rapports qui imposait le passage rapide aux modèles alors que l'étude de la base de données méritait que l'on s'attarde plus sur les données, pour créer de nouvelles variables plus influentes.
- Jeu de données :
 - En 2018, les modalités de recueil de la variable cible (gravité) par les forces de l'ordre ont changé: nous avons donc dû restreindre notre analyse aux données recueillies à partir de 2019.
- Compétences techniques / théoriques :
 - Lors de la modélisation des séries temporelles nécessitant une double saisonnalité (annuelle et hebdomadaire), nous avons dû trouver des modèles adéquats. De plus, la modélisation LSTM (série temporelle utilisant le Deep Learning) a nécessité le déblocage du cours correspondant afin de comprendre l'implémentation du code pour réaliser les prévisions sur les dates futures.
 - Le traitement spécifique des données géographiques aurait été un plus dans cette analyse des accidents sur le territoire, car nous disposions des coordonnées précises de chaque accident (longitude / latitude). Nous les avons entrées dans les modèles comme des variables continues classiques, alors qu'une modélisation géostatistique appropriée aurait sûrement permis d'en tirer plus d'information.
- Pertinence des données à disposition:
 - La base de données disponible en open data est épurée des données "spécifiques relatives aux usagers et aux véhicules et à leur comportement dans la mesure où la divulgation de ces données porterait atteinte à la protection de la vie privée des personnes physiques aisément identifiables ou ferait apparaître le comportement de telles personnes alors que la divulgation de ce comportement pourrait leur porter préjudice (avis de la CADA – 2 janvier 2012)" : parmi celles-ci pourraient se trouver des informations prédictives de la gravité (consommation d'alcools ou de drogues, vitesse excessive etc).
- Enjeux IT :

- Le jeu de données créé par la concaténation des données issues de data.gouv.fr était trop volumineux pour être stocké directement sur GitHub. Nous avons donc choisi de le stocker localement sur nos ordinateurs personnels et de le charger en début de notebook afin de pouvoir l'utiliser lors des modélisations. A cette fin, conformément aux bonnes pratiques de programmation, nous avons indiqué l'emplacement local propre à chacun dans un fichier de configuration personnel, non versionné. Nous avons rencontré le même problème lors de l'enregistrement de certains modèles de Machine ou Deep Learning sur GitHub. Nous avons résolu le problème en utilisant une compression des fichiers de modèles au format zip.

VII.3 Bilan

Le projet a été mené par une équipe de 4 personnes dont voici la répartition des tâches :

	Matthieu Claudel	Vanessa Ibert	Camille Pelat	Nadège Reboul
Preprocessing et visualisation				
Agrégation d'un dataset à partir des sources	x	x	x	x
Exploration du dataset	x	x	x	x
Preprocessing dataset	x	x	x	x
Data Vizualisation	x	x	x	x
Modélisation - Séries temporelles				
Tendance - Saisonnalité - Bruit		x		
Baselines		x		
SARIMAX + Exog Fourier		x		x
MSTL		x		
PROPHET		x		
LSTM		x		
Modélisation - Machine Learning				
Régression logistique				x
LinearSVC + Nystrom				x
Decision Tree - Random Forest - Balanced		x		
CatBoost				x
XGBoost	x			
KNN			x	
Modélisation - Deep Learning				
Keras (class_weight)	x	x		
Keras (undersampling)		x		
Keras (oversampling)		x		
PyTorch				x
TabNet				x
Amélioration des résultats				
Classification binaire		x		x
Ajout de la variable 'catv_percute'		x		
Ajout de la variable 'nb_usager_gr'		x	x	

Les différents tâches effectuées lors de ce projet peuvent s'inscrire dans les process métiers suivants :

- Data Analyst : pour la préparation des données et leur visualisation
- Data Scientist : pour les modélisations et optimisations
- Data Engineer : pour l'acquisition des données et la mise en place de pipelines
- MLOps : pour la présentation sur Streamlit

VII.4 Perspectives

Afin d'optimiser les performances du modèle, nous avons essayé 3 pistes de manière indépendante qui permettent chacune d'obtenir de meilleurs résultats tout en conservant de bons f1-scores pour la prédictions des modalités tués et blessés hospitalisés :

- augmentation de 0,14% de l'accuracy par ajout de la variable 'catv_percute' et modification des variables 'catv', 'obs' et 'obsm' (mais en ayant les données de 'catv_percute' que pour les piétons soit 7,62% du jeu de données),
- augmentation de 0,45% de l'accuracy par ajout de la variable 'nb_usagers_gr',
- augmentation moyenne de 2% de l'accuracy pour chaque classification binaire lorsque l'on sépare le jeu de données en 4 pour réaliser des classifications binaires.

Nous pensons qu'en combinant ces approches, cela pourrait encore améliorer les prédictions.

La source de la performance moyenne vient en majeure partie du dataset et notamment des données disponibles. Une étude plus approfondie de ces variables permettrait d'envisager soit d'agréger de nouvelles sources de données complémentaires soit de revoir l'utilisation de certaines variables dans la modélisation (par exemple, introduire la variable vitesse au moment de l'accident...)

BIBLIOGRAPHIE

- Barmoudeh, L., Baghislani, H., & Martino, S. (2022). Bayesian spatial analysis of crash severity data with the INLA approach: Assessment of different identification constraints. *Accident Analysis & Prevention*. doi:<https://doi.org/10.1016/j.aap.2022.106570>
- Bhardwaj, C. A., Mishra, M., & Desikan, K. (s.d.). *Dynamic Feature Scaling for K-Nearest Neighbor Algorithm*. Récupéré sur <https://arxiv.org/pdf/1811.05062.pdf>
- CEREMA. (2024, 02 11). *Evolution de la mortalité sur la période 1968-2022*. Récupéré sur <https://dataviz.cerema.fr/securite-routiere-series/>
- Effati, M., & Sadeghi-Niaraki, A. (2015). A semantic-based classification and regression tree approach for modelling complex spatial rules in motor vehicle crashes domain. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. doi:<https://doi.org/10.1002/widm.1152>
- Kaleko, D. (2017, 10 30). *Feature Engineering - Handling Cyclical Features*. Consulté le 02 11, 2024, sur <https://blog.davidkaleko.com/feature-engineering-cyclical-features.html>
- Lahlou Mimi, A. (2018). *Accidents in France from 2005 to 2016, Kaggle*. Récupéré sur Kaggle: <https://www.kaggle.com/datasets/ahmedlahlou/accidents-in-france-from-2005-to-2016>
- Maxime, m. (2019). *Predict Severity of Accidents, Kaggle*.
- Ministère de la transition écologique et de la cohésion des territoires. (2024, 02 11). *Données et études statistiques*. Récupéré sur <https://www.statistiques.developpement-durable.gouv.fr/>
- Ministère de l'Intérieur et des Outre-Mer. (2013, màj 2023). *Bases de données annuelles des accidents corporels de la circulation routière - Années de 2005 à 2022*. Consulté le 02 2024, sur data.gouv.fr/.
- Observatoire national interministériel de la sécurité routière. (2024, 02 11). Récupéré sur <https://www.onisr.securite-routiere.gouv.fr/>
- Routes de France. (2023). *Etat de la route - 2023*. Récupéré sur <https://www.routesdefrance.com/wp-content/uploads/2023/07/rdf-edlr-2023.pdf>
- Statista Research Department. (2024, 02 11). *Statista*. Récupéré sur <https://fr.statista.com/statistiques/943831/moyens-transport-utilises-deplacements-quotidiens-france/>
- Talbi, I. (2020, 09 6). *KGBoost vs Random Forest : prédire la gravité d'un accident de la route*. Récupéré sur La revue IA: <https://larevueia.fr/xgboost-vs-random-forest-predire-la-gravite-dun-accident-de-la-route/>