# Rapport mi-projet
# Fairness pour l'IA

Erwan LEMATTRE, Yannis CHUPIN

6 mars 2024

# Table des matières

# 1 Introduction

Ce projet a pour objectif d'analyser les accidents de la circulation routière afin de pouvoir dire à partir des données d'un véhicule accidenté si l'accident est mortel ou non. Les données sont des données libres mises à disposition par le *Ministère de l'intérieur et des Outre-Mer*. Le jeu de donnée correspond aux accidents de 2005 à 2022 en France. Nous allons dans une première partie analyser ces données afin d'extraire les informations utiles à l'apprentissage et de pouvoir repérer d'éventuelles sources de biais pour notre modèle.

Vous pouvez retrouver le code sur le GitHub du projet. Le fichier `main.ipynb` contient le code principale que nous allons suivre tout au long de ce rapport. Le fichier `utils.py` contient toutes les fonctions auxilières que nous utilisons dans le fichier principale.

# 2 Découverte du jeu de données

## 2.1 La base de données

La base de données est composée de plusieurs tables : *usagers*, *vehicules*, *lieux* et *caracteristiques*. Nous avons joint ces quatres parties pour obtenir un dataframe contenant une cinquantaine de colonnes. On peut retrouver l'ensemble des attributs dans l'annexe A

## 2.2 Répartition des données

Afin de pouvoir conserver les données utiles pour l'apprentissage nous avons analysé la répartition des différentes données dans notre dataframe. Nous avons ainsi pu faire différentes observations.
Voici quelques-unes d'entre elles qui nous sont ensuite utiles pour la préparation des données.

### 2.2.1 Catégories de véhicule

La base de données nous donne beaucoup de catégories différentes. Nous avons cependant pu remarquer que la majorité des véhicules sont dans seulement 5 catégories.

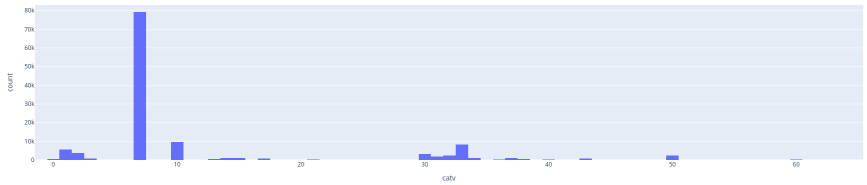



FIGURE 1 – Répartition des catégories de véhicules

### 2.2.2 Gravité de l'accident

En affichant l'effectif d'accidents mortels nous avons pu remarquer qu'ils ne représentent qu'une infime partie des accidents. Le peu de données sur ces accidents ne nous permet pas l'apprentissage d'un modèle. C'est la raison pour laquelle nous avons décidé de nous intéresser non pas à la mortalité à l'échelle d'une personne mais plutôt à l'échelle d'un accident. Nous nous mettons pour cela au niveau d'un véhiule car cela nous permet de conserver plus d'informations

(à l'échelle d'un accident on aurait dû enlever trop d'informations pour conserver seulement les attributs plus généraux à l'accident).

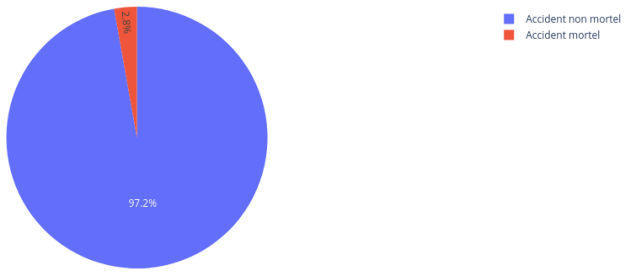



FIGURE 2 – Proportion d'accidents mortels

# 3 Préparation des données

À partir des observations précédentes, nous avons supprimé les attributs moins intéressant pour l'apprentissage et nous avons modifié certains attributs afin d'en extraire les informations intéressantes.
Les attributs supprimés sont : *voie*, *v1*, *v2*, *pr*, *pr1*, *lartpc*, *larrout*, *num_veh*, *occutc*, *adr*, *senc*, *etatp*, *actp*, *manv*, *jour*, *com*, *hrmn*, *motor*, *place*, *vosp*, *locp*.

Nous avons effectué les modifications suivantes :

- Création d'un attribut *mortal* qui vaut 1 si le véhicule est impliqué dans un accident mortel, 0 sinon.
- À partir de l'attribut *sexe* nous avons créé un attribut *sexe_conducteur* qui garde seulement le sexe du conducteur du véhicule.
- Création d'un attribut *piéton* qui vaut 1 si un piéton est impliqué dans l'accident, sinon 0.
- Nous avons utilisé l'année de naissance et l'année de l'accident pour récupérer l'âge du conducteur.
- L'attribut *vma* a été découpé en 4 catégories de vitesse.
- Pour les attributs *catv* et *vatr* nous avons gardé les valeurs le plus représentées dans la base de données.

Nous avons également reduit les valeurs de certains attributs. Par exemple pour des attributs avec des valeurs telles que *Non-renseigné*, *Autre* ... Nous avons regroupé ces valeurs en une seule valeur. L'objectif était ici de simplifier en réduisant les catégories mais également d'améliorer les performances de notre modèle.

# 4 Analyse des données

Une fois nos données préparées, nous avons pu les visualiser. Nous allons montrer dans les deux prochaines parties les observations intéressantes que nous avons pu faire lors de l'analyse de

notre dataset.

## 4.1 Analyse univarié des données

### 4.1.1 Les accidents mortels

Une donnée intéressante à observer est la proportion de véhicules impliqués dans un accident mortel. C'est en effet la valeur que nous voulons prédire.

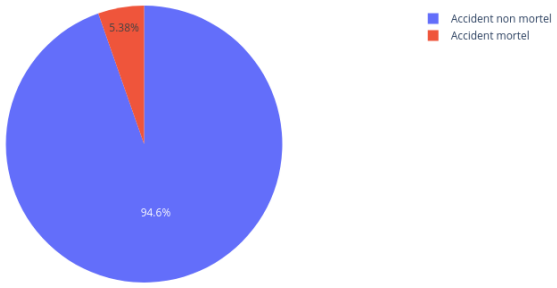



FIGURE 3 – Proportion des véhicules impliqués dans un accident mortel

Nous pouvons remarquer sur la figure 3 que le fait de s'intéresser aux véhcules impliqués dans un accident mortel et non plus aux personnes victimes permet de doubler le pourcentage. Même si cette proportion reste faible, cela va nous permettre d'avoir plus de données dans la catégorie mortel lors de l'apprentissage et par conséquent d'avoir un meilleur modèle.

### 4.1.2 Les piétons

Nous nous sommes ensuite intéressé aux accidents dans lesquels un piéton est impliqué. La figure 4 nous montre qu'un peu moins de 10% des accidents impliquent un piéton.

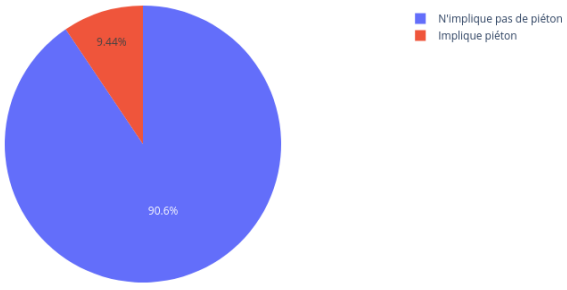



FIGURE 4 – Proportion des accidents avec piéton

### 4.1.3 L'âge

Nous pouvons visualiser l'âge des conducteurs via une boîte à moustache. La figure 5 nous montre la répartition de l'âge des conducteurs. Lors du prétraitement des données, les valeurs aberrantes ont été enlevée. On retrouve donc logiquement des âges contenus entre 0 et 100 ans. L'âge médian des conducteurs est 33 ans avec le premier quatile à 21 et le troisième quatile à 49 ans. Même si on peut imaginer que des valeurs sont fausse (il y a des conducteurs de moins de 16 ans), les valeurs sont tout de même assez cohérentes par rapport à ce que l'on pourrait imaginer de la répartition de l'âge des conducteurs.

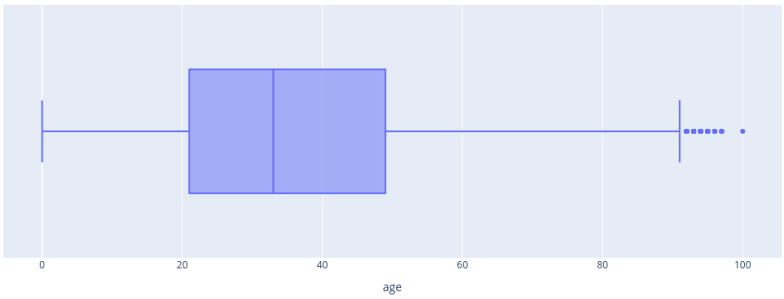



FIGURE 5 – Âge des conducteurs

### 4.1.4 Le genre des conducteurs

Le genre des conducteurs est assez intéressant à analyser. Sur la figure 6 nous pouvons remarquer différence importante entre le nombre de femme au volant (indice 0) et le nombre d'hommes au volant (indice 1). Cette différence pourrait être une source de biais pour notre modèle. En effet le fait qu'il y ait beaucoup plus de données d'accident avec des hommes ne signifie pas qu'il y a plus de chance d'avoir un accident si on est un homme. Cela signifie peut-être que la proportion d'homme au volant est plus élévée et donc qu'il y a plus d'accident avec un homme au volant car il y a plus d'homme au volant. Le risque ici est que notre modèle associe une homme à un accident mortel car il y a beaucoup plus d'accidents mortels avec un homme au volant.

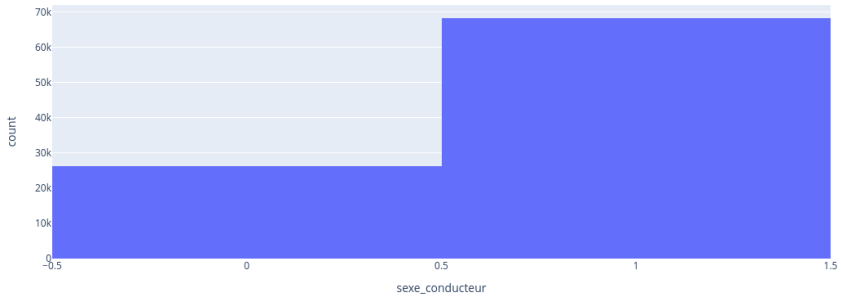



FIGURE 6 – Genre des conducteurs

### 4.1.5 Le type de collision

La figure 7 montre la répartition des différents types de collisions dans notre dataset. On peut remarquer que tous les types de collisions sont plutôt bien représentés dans notre dataset. C'est un attribut qui pourra être assez intéressant pour l'apprentissage.

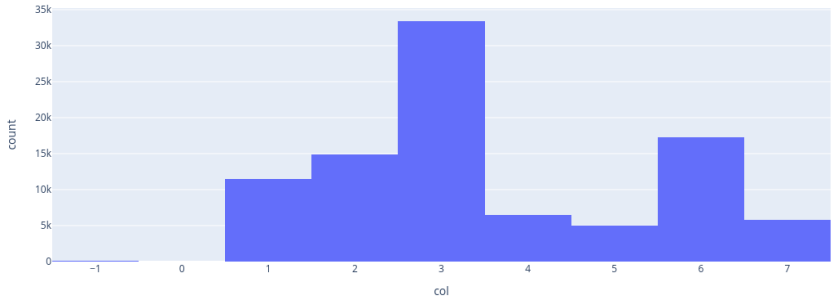



FIGURE 7 – Les types de collision

## 4.2 Analyse bivariée des données

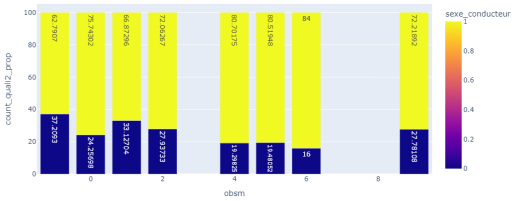
Nous allons dans cette partie donner quelques exemples intéressant obtenus lors de l'analyse bivariée. On peut retrouver l'ensemble des graphiques observés dans le fichier `main.ipynb`.

### 4.2.1 Le genre du conducteur

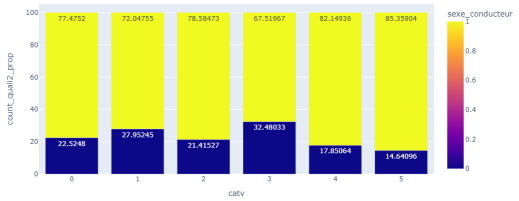
Un attribut qu'il est intéressant d'analyser est le genre du conducteur. En effet il peut être source de biais s'il y a un déséquilibre entre homme et femme. On retrouve globalement la même proportion dans la corrélation que l'on soit homme ou femme. Les hommes étant beaucoup plus représentés dans le dataset, la proportion d'hommes est logiquement plus élevée. On peut cependant faire quelques remarques.

La figure 8a nous montre une proportion de femme moins élevée quand *obsm* vaut *6*. La proportion de femmes est deux fois plus élevée quand *obsm* vaut *1*. Ceci pourrait biaiser notre modèle.

Sur la figure 8b, on remarque également une proportion différente de femmes en fonction du type de véhicule. On pourrait expliquer cela par le fait que certains véhicules sont dans la réalité plus utilisés par les hommes, par exemple pourrait imaginer qu'il y a plus d'hommes qui conduisent des motos. Il faudra être vigilant car cela peut être source de biais. Il se peut que notre modèle associe une moto à un homme. Et que dans le cas d'une femme sur une moto le résultat soit soit forcément un accident mortel, soit forcément un accident non mortel.



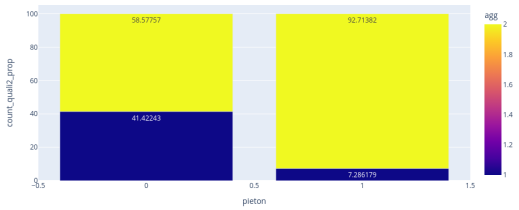

(a) Proportion femmes/hommes en fonction de la présence d'un obstacle dans l'accident



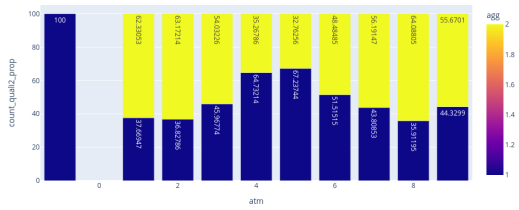

(b) Proportion femmes/hommes en fonction de la catégorie du véhicule

### 4.2.2 Agglomération

L'attribut *agg* indique si l'accident a eu lieu en agglomération ou hors agglomération. On retrouve des corrélations plutôt logiques avec un certain nombre d'attributs comme par exemple une proportion d'accident avec piéton plus élévé en agglomération (figure 9a). On peut également remarquer (figure 9b) que la proportion d'accident en agglomération varie avec les conditions atmosphériques. Toutes ces corrélations peuvent nous indiquer que cet attribut va pouvoir contribuer à nos prédictions.




(a) Proportion d'accidents en agglomération avec piéton et sans piéton




(b) Proportion de d'accident en agglomération en fonction des conditions atmosphériques

# 5 Apprentissage

## 5.1 Split

Séparer nos données en deux sets demandait de tenir compte d'une spécéificité. Nous nous plaçons du point de vue d'un véhicule. Mais c'est bien souvent plusieurs véhiécules qui sont impliqués dans un accident. Il nous a donc fallut adapter le split pour que les véhicule d'un même accident soit dans le même set.

Par ailleurs, nous avons pu constater que ne pas adopter cette approche, provoquait un overfit lors du test. Typiquement, un trop grand nombre de mort était trouvés avec succès.

## 5.2 One Hot Encoding

La base de donnée nous fournit pour la plus part des attributs des données qui sont convertibles en entiers. Par exemple une catégorie de véhicule (*catv*) est dénomée par un chiffre. L'attribut cependant reste catégoriel. Dans les faits, Le seul attribut quantitatif qu'il nous reste est l'âge du conducteur. Le reste est soit binaire, soit qualitatif.

Tous les attributs qualitatifs ont donc été transformés en OneHot. Au départ nous souhaitions optimiser le nombre de colonnes en faisant les one hot par nous même. Typiquement, preonons surf (l'état de la surface de la route). Il est inutile de créer une colonne Non-renseigné, Autre ou normale. Nous ne sommes intéressés que par les états spécifiques de la route.

Cet approche cependant était source d'ennuis puisqu'elle empêchait la réalisation de l'audit. Nous avons donc utilisés la méthode standard pour nos oneHot.

## 5.3 Random Tree Classifier

Une fois les données prêtes, nous avons utilisés le classifieur d'arbres aléatoires pour obtenir un modèle. C'est le premier que nous avons utilisé. Pour ce qui est des résultats, l'accuracy score est de *1* pour l'entrainement et de *0.9014* pour le teste.

Pour ce qui est de la matrice de confusion, on remarque que *27687* véhcules ont été classés à raison en tant que accidents non létaux. Tandis que *291* accidents mortels ont été trouvés. Toutefois *1413* accidents mortel ont été resencés à tord comme étant non létaux et *1648* véhicule ont subit le sort inverse.


FIGURE 10 – La matrice de confusion pour le random tree classifier

En résumé, on remaque que le modèle réussit très bien à classifier les accidents qui n'ont pas conduit à la mort de quelqu'un. Cemendant, il lui apparaît bien plus difficile de trouver les véhicules impliqués accidents qui sont mortels. On peut attribuer ceci au fait que ces accidents létaux ne représentent que *5.38%* des accidents rétertoriés dans la base.

## 5.4 GaussianNB

Avancés dans le projet, nous avons voulus tester plusieurs classifieurs, tous ont donné des résultats différents, pour certains médiocres. Mais nous avons trouver un autre classifieur qui donnait des résultats intéressants : GaussianNB.

Il ne maximise pas notre accuracy score puisqu'il n'est que de *0.8734* à l'entrainement et de *0.8677* pour les tests. Ce qui a toutefois retenu notre attention c'est le nombre d'accidents létaux qu'il parvient à détecter : *508*. Réduisant ainsi l'erreur des accidents mortel resencés à tord comme étant non létaux à seulement *1196*. Ceci entrainne malheureusement un perte de performances pour ce qui est de la détection d'accidents non mortel comme on peut le voir ci-dessous.

Un autre avantage qui peut ressembler à un inconvienent au départ est que ce classifieur demande des nombres et non des oneHot. Or nous en avons à foison. En dépis de ça, nos valeurs au départ


FIGURE 11 – La matrice de confusion avec GaussianNB

sont en format numérique, rendant donc nos données compatibles avec cette approche. Ceci nous ouvre alors la possibilité de tester d'autres choses puis que le classifieur donne des probabilités et non des arbres. Au premier rang desquels *xplique*.

Par soucis de concision cependant, nous prendrons le modèle issus de *random tree classifier* pour tous les audits qui vont suivre, puisque c'est le premier avec lequel nous avons travaillé. On notera tout de même que lorsqu'on effectue ces audits avec *GaussianNB* Les résulats diffèrent en bien des points de ceux observés avec *random tree classifier*, preuve qu'ils ont une approche bien différente.

## 5.5 Base Rate

Nous avons voulu mesurer les résultats sur deux données à protéger potentiellement : le fait qu'un piéton soit impliqué dans un accident et le sexe du conducteur. Voici les résultats :

sexe_conducteur 1 (un homme)

disparate_impact 1.21767750298587 1.6331199049428085

P_rule_disparate_impact 0.8212355057458954 0.6123249107266375

demography_parity 0.011747835864606336 0.02383701239031672

Ce qui est flagrant ici, c'est qu'il existe un disparité dans la prédiction d'accident mortel pour les hommes. Allant par ailleur dans le sens de ce qu'indique disparate impact par la suite. Notons cependant que la disparité démographique existe mais est très faible.

pieton 1 (un piéton est impliqué)

disparate_impact 1.0538191011165416 1.0968113720110815

P_rule_disparate_impact 0.9489294689576994 0.9117337999207913

demography_parity 0.003345154811069756 0.005266913173450724

Pour ce qui est de l'implication d'un piéton dans un accident, on constate ici aussi une légère disparité, soulignant leur plus grande implication dans des accidents mortels. Ceci concorde encore une fois avec le *disparaite_impact*. Démographiquement, le cas est similaire au sexe, puisqu'on constate une légère disparité en leur faveur. Bien que cette fois la disparité soit plus faible encore.

# 6 Audit du modèle

## 6.1 Génération des contrefactuels avec Dice

Dice nous a permis d'établir quels attributs influent sur la prédiction de notre modèle. Les résultats sont cependant assez variants d'une exécution à l'autre. On peut tout de même retrouver lesquels sont fréquemment impliqués dans le changement des exemples contrefactuels. Parmi eux on peut noter que *âge* est souvent représenté. Le taux d'accidents (notamment mortels) étant plus élevé chez les jeunes et les personnes âgées, cela paraît assez cohérent. L'attribut *obsm* est également souvent représenté dans les exemples contrefactuels. On a pu remarquer que l'attribut *col* est peu représenté dans les résultats contrairement à l'hypothèse que nous avions pu faire lors de l'analyse.

Enfin les attributs *dep* et *sexe_ conducteur* sont également représentés. Ils pourraient être sources de biais, notamment l'attribut *dep*, le modèle pourrait associer un département à une prédiction d'accident mortel.

## 6.2 BlackBoxAuditing

Les résultats que nous allonjs analysés sont issus du modèle généré par *random tree classifier*.

Tout d'abord l'audit a porté sur les 24 features que nous avons conservés afin d'élaborer ce modèle. Ce qui marque avec cet audit provient du sommaire. On y constate le role pédominant de l'age pour l'accuracy score. À 0.84, ce dernier éclipse tous les autres. Une explication pourrait venir de ce que nous avons constaté dans notre analyse univariée : il ya une surreprésentation des jeunes de 20-21 ans dans ce set. Ceci est en réalité loin de surprendre puisqu'il s'aligne avec la politique de prix pratiqué par les assureurs envers les jeunes conducteurs.

Par la suite on peut s'intéresser à l'évolution de l'accuracy en fonction du niveau de réparation appliqué à chaque attribut :

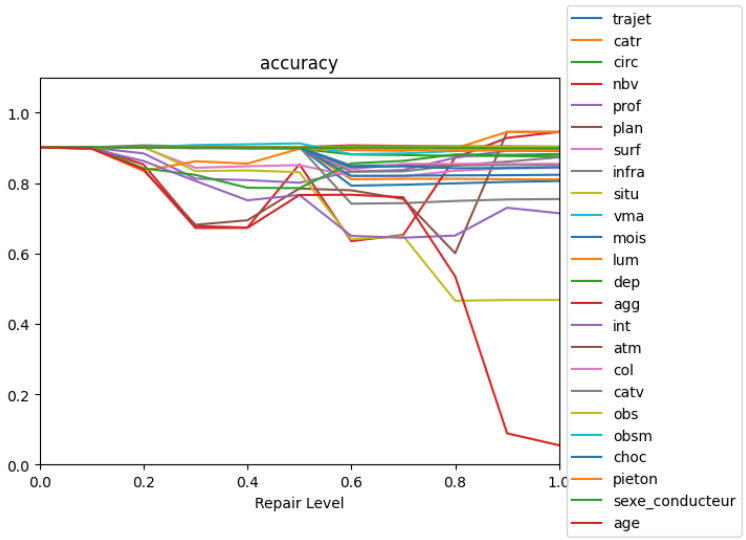



FIGURE 12 – L'accuracy en fonction du niveau de réparation pour chaque attribut.

Ce qui est marquant ici est la disparité de réaction lors de l'application de mesures de réparations.

On reconnait les attributs dont l'accuracy est au plus haut en regardant ceux qui s'effondrent le plus. Au premier rang desquels l'age s'illustre à nouveau. À la lumière de ceci, il apparait évident que modèle dicrimine fortement les véhicule en fonction de l'âge du conducteur. Dans une moindre mesure on constate la même chose pour *situ*, qui fait référence à la localisation géographique de l'accident. Ce qui parrait sencé, étant donné que certains lieu sont plus dangereux que d'autres. Enfin, notons que le type d'intersection et la catégorie du véhicule est elle aussi source de discriminations de la part de notre modèle. Encore une fois cela semble faire sense.

Toutefois, notons l'absence de quelques attributs notables tels que le nombre de voies, le sexe du conducteur (chose intéressante : ce n'est pas le cas avec GaussianNB), le type de colision, le type d'obstacle heutré ou encore le fait qu'un piéton soit impliqué. Pour ce qui est des piétons, ça peut parraitre étonant, on peut émettre l'hypothèse que leur présence est souvent en ville là où les vitesses sont basses. entrainant donc moins de mort.

Maintenant regardons l'équilibre du modèle à l'aide du BCR :

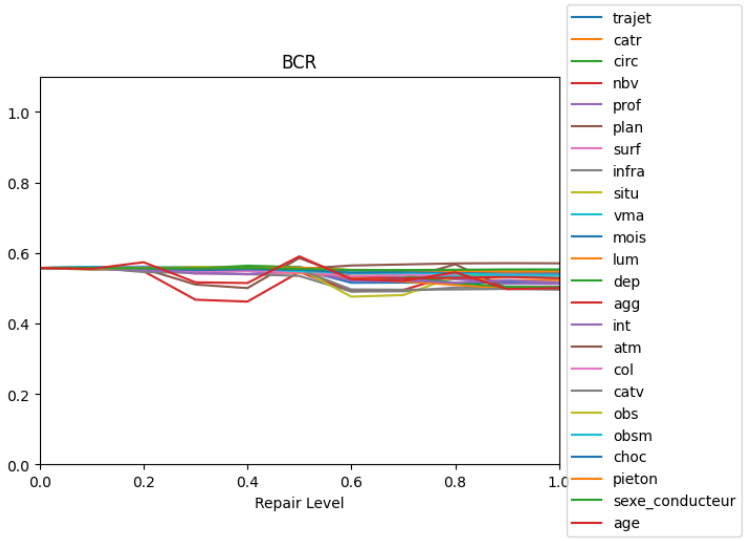



FIGURE 13 – Le BCR en fonction du niveau de réparation pour chaque attribut.

On remarque ici que les attributs sont plutot centrés autour de 0.5 que la plus part reste rectiligne, mais que certaint divergent quelque peu. Cet évasement est dû à des attributs déjà connus tels que âge ou catv. D'autres influent aussi mais ne s'étaient pas illustré précédament à l'image du nombre de voies, du profile de la route ainsi que la luninosité.

Malgré cela, l'évasement assez faible laisse dire que le modèle est plutot équilibré. De plus, comme mentionné précédament dans la section dédié au split, nous avons pu tester le cas où l'on acceptait que des véhicules impliqués dans le même accident se retrouvent dans des sets différents (entrainement / test). Ce qu'il ressortait alors du BCR était un évasement beaucoup plus important et une origine située vers 0.68. Ce qui fait dire que cette version était bel et bien trop modelée sur le train set, l'équilibre était rompu.

## 6.3 Expliquer le modèle avec les valeurs de Shapley

Afin d'analyser la contribution des différents attributs dans notre modèle, nous avons utilisé les valeurs de Shapley. Nous avons d'abord essayé de calculer la valeur exacte avec la fonction `ShapleyValues` du module *ShapKit*. Cependant le nombre d'attributs de notre dataset est trop important et le calcul trop long. Il a donc fallu calculer une approximation de la valeur de Shapley avec la fonction `MonteCarloShapley`. Nous devons cependant être prudents car le calcul de l'approximation nous donne une tendance pour notre modèle mais certainement pas une valeur exacte. On peut le constater en changeant le nombre d'itérations qui produit un résultat différent.

Avec `n_iter=1000` on obtient le diagramme en cascade de la figure 14. Ce diagramme nous montre en partant d'un score référence de 1 quels attributs contribuent à obtenir un score de 0 (l'utilisation d'un arbre de décision nous donne forcément une valeur binaire 0 ou 1 d'où le passage de 1 à 0).

Les attributs qui contribuent le plus sont ici *trajet*, *agg*, *circ*, *mois* et *obsm*. On aurait pu ici s'attendre à plus de contribution de *lum*, *vma* ou encore *col* (qui ici a une contribution mineure).

Une donnée intéressante ici est la valeur pour l'attribut *sexe_conducteur*. On nous donne une contribution nulle de cet attribut ce qui peut nous encourager dans l'idée que le sexe n'est pas un biais pour notre modèle. Ce résultat étant une approximation, il est possible que la véritable valeur ne soit pas complètement nulle mais du moins assez faible.

Enfin, ce résultat nous montre que seule une petite partie des attributs contribue au résultat. On aurait voulu une contribution un peu plus répartie entre les attributs car ici le changement d'une petite partie des attributs pourrait faire basculer le résultat or on sait bien que les 5 attributs qui contribuent ici le plus ne permettent pas à eux seuls d'expliquer ce changement.

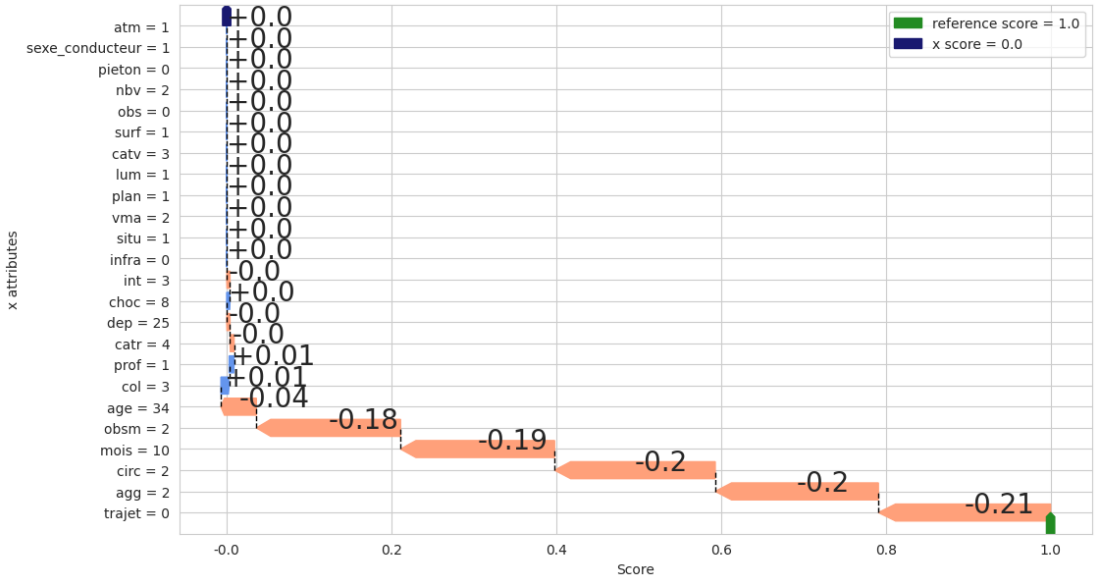



FIGURE 14 – Une approximation des valeurs de Shapley

### 6.4 Expliquer le modèle avec Lime

Nous avons également utilisé *Lime* afin d'avoir une idée des attributs qui contribuent le plus aux prédictions de notre modèle. On peut retrouver le code correspondant dans le fichier `lime.ipynb`.

Nous avons d'abord récupéré les indices des prédictions positives de notre modèle (avec quelles entrées notre modèle prédit 1 ?). Ainsi on a pu regarder dans les cas où les prédictions sont 0 et les cas où les prédictions sont 1. On peut remarquer qu'on trouve des variables similaires à celles trouvées avec les valeurs de Shapley. On retrouve notamment assez fréquemment l'attribut *agg*. Cependant, contrairement au résultat que pouvait nous donner *ShapKit* l'attribut *mois* est moins représenté. Les contributions sont également plus réparties entre les attributs. Enfin il semble que comme nous avions pu l'observer avec les valeurs de Shapley l'attribut *sexe_conducteur* ait un faible impact sur la prédiction.

## 7 Conclusion

# A Les données de notre dataset

| Attribut | Description |
|---|---|
| *Num_ Acc* | Numéro d'identifiant de l'accident |
| *jour mois* | Jour de l'accident, mois de l'accident |
| *an* | Année de l'accident. |
| *hrmn* | Heure et minutes de l'accident. |
| *lum* | Lumière : conditions d'éclairage dans lesquelles l'accident s'est produit. |
| *dep* | Département : Code INSEE. |
| *com* | Commune : Le numéro de commune est un code donné par l'INSEE. |
| *agg* | Localisation dans une aglomération ou non. |
| *int* | Si intersection : type d'intersection. |
| *atm* | Conditions atmosphériques. |
| *col* | Type de collision. |
| *adr* | Adresse postale : variable renseignée pour les accidents survenus en agglomération. |
| *lat* | Latitude. |
| *long* | Longitude. |
| *catr* | Catégorie de route (type de route). |
| *voie* | Numéro de la route. |
| *circ* | Régime de circulation. |
| *nbv* | Nombre total de voies de circulation. |
| *vosp* | Signale l'existence d'une voie réservée. |
| *prof* | Profil en long décrit la déclivité de la route à l'endroit de l'accident. |
| *plan* | Tracé en plan de la route. |
| *surf* | Etat de la surface de la route. |
| *infra* | Aménagement — Infrastructure s'il y en a. |
| *situ* | Situation géographique de l'accident. |
| *vma* | Numéro d'identifiant de l'accident. |
| *id_ vehicule* | Identifiant du véhicule, foreign key. |
| *catv* | Catégorie du véhicule accidentée. |
| *obs* | Type d'obstacle heurté. |
| *obsm* | Type d'obstacle mobile heurté. |
| *choc* | Point de choc initial. |
| *manv* | Manoeuvre principale avant l'accident. |
| *catu* | Catégorie d'usager (Conducteur, passager, piéton) |
| *grav* | Gravité de l'accident pour l'usager. |
| *sexe* | Sexe du conducteur de la voiture (calculé dans le notebook). |
| *trajet* | Motif du déplacement au moment de l'accident. |
| *mortal* | Insique si le véhicule est impliqué dans un accident mortel. Attribut calculé dans le notebook. |