

Rapport mi-projet Fairness pour l'IA

Erwan LEMATTRE, Yannis CHUPIN

6 mars 2024

Table des matières

1	Introduction	3
2	Découverte du jeu de données	3
2.1	La base de données	3
2.2	Répartition des données	3
2.2.1	Catégories de véhicule	3
2.2.2	Gravité de l'accident	3
3	Préparation des données	4
4	Analyse des données	4
4.1	Analyse univariée des données	5
4.1.1	Les accidents mortels	5
4.1.2	Les piétons	5
4.1.3	L'âge	6
4.1.4	Le genre des conducteurs	6
4.1.5	Le type de collision	7
4.2	Analyse bivariée des données	7
4.2.1	Le genre du conducteur	7
4.2.2	L'âge du conducteur	8
5	Apprentissage	8
5.1	Split	8
5.2	Encodage One Hot	8
5.3	Arbre de décision	9
5.4	GaussianNB	9
5.5	Base Rate	10
6	Audit du modèle	11
6.1	Génération des contrefactuels avec Dice	11
6.2	BlackBoxAuditing	11
6.3	Expliquer le modèle avec les valeurs de Shapley	13
6.4	Expliquer le modèle avec Lime	14
7	Conclusion	14
A	Les données de notre dataset	15

1 Introduction

Ce projet a pour objectif d’analyser les accidents de la circulation routière afin de pouvoir déterminer 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 données 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. Puis nous essaierons de repérer d’éventuelles sources de biais affectant notre modèle.

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

2 Découverte du jeu de données

2.1 La base de données

La base de données est composée de plusieurs tables : *usagers*, *véhicules*, *lieux* et *caractéristiques*. Nous avons joint ces quatre 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 contient beaucoup de type de véhicules différents. Nous avons cependant pu remarquer que la majorité des véhicules sont dans seulement 5 catégories.

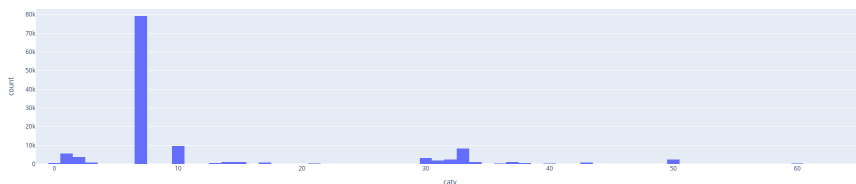


FIGURE 1 – Répartition des catégories de véhicules

2.2.2 Gravité de l’accident

En affichant l’effectif d’individus décédés dans un accident, nous avons pu remarquer qu’ils ne représentent qu’une infime partie des individus accidentés. Leur proportion est si faible que ça ne nous permet pas d’apprendre 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 véhicule car cela nous permet de conserver

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

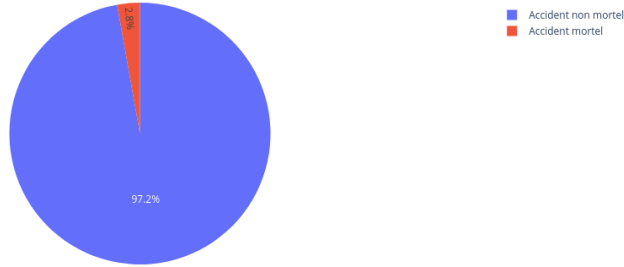


FIGURE 2 – Proportion d'accidents mortels

3 Préparation des données

À partir des observations précédentes, nous avons supprimé les attributs moins intéressants 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*, *larroul*, *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 *catr*, nous avons gardé les valeurs les plus représentées dans la base de données.

Nous avons également réduit 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 univariée 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.

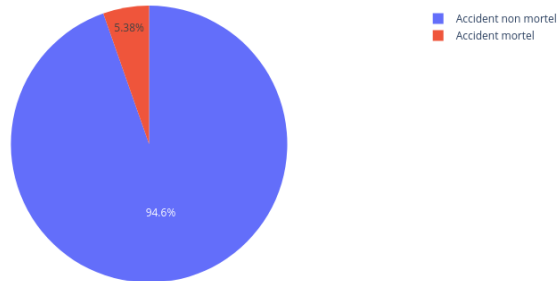


FIGURE 3 – Proportion des véhicules impliqués dans un accident mortel

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

4.1.2 Les piétons

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

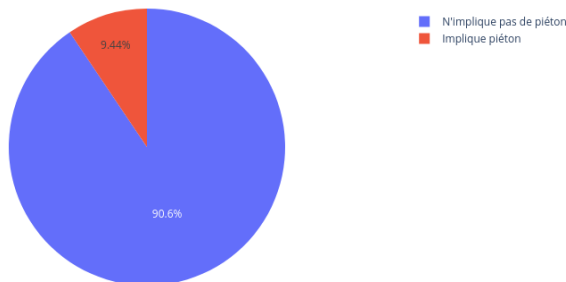


FIGURE 4 – Proportion des accidents avec piéton

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 enlevées. On retrouve donc logiquement des âges contenus entre 0 et 100 ans. L'âge médian des conducteurs est 33 ans avec le premier quartile à 21 et le troisième quartile à 49 ans. Même si on peut imaginer que des valeurs sont fausses (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.

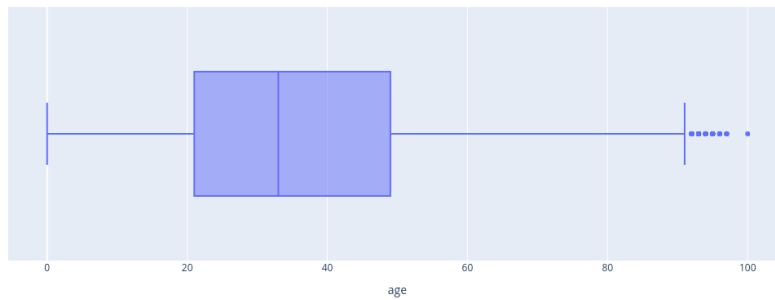


FIGURE 5 – Âge des conducteurs

4.1.4 Le genre des conducteurs

Le genre des conducteurs est assez intéressant à analyser. Sur la figure 6, nous pouvons remarquer une différence importante entre le nombre de femmes au volant d'un véhicule ayant eu un accident (indice 0) et le nombre d'hommes (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 chances d'avoir un accident si on est un homme. Cela signifie peut-être que la proportion d'hommes au volant est plus élevée et donc qu'il y a plus d'accidents avec un homme au volant car il y a plus d'hommes au volant. Le risque ici est que notre modèle associe un homme à un accident mortel car il y a beaucoup plus d'accidents mortels avec un homme au volant.

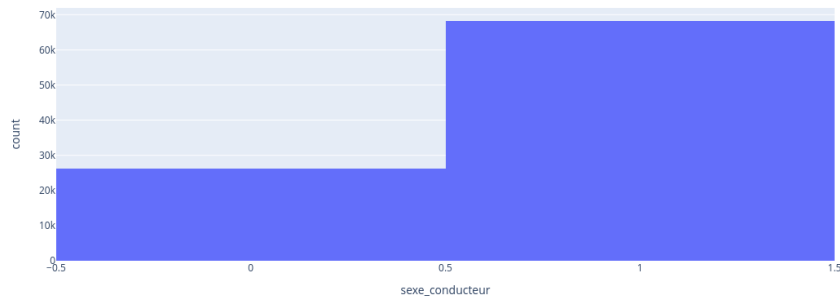


FIGURE 6 – Genre des conducteurs

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.

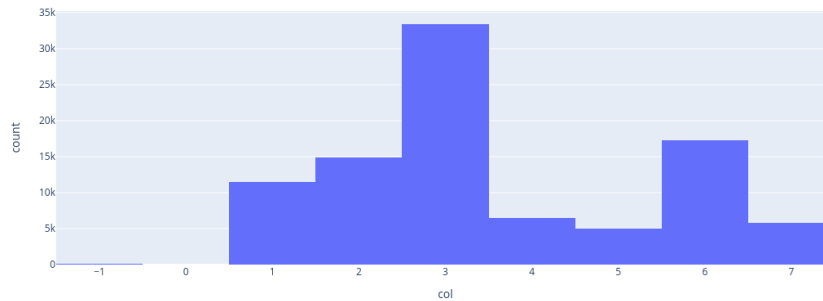


FIGURE 7 – Les types de collision

4.2 Analyse bivariée des données

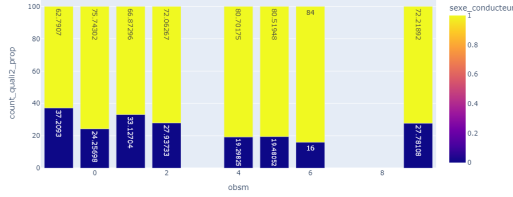
Nous allons dans cette partie donner quelques exemples intéressants obtenus lors de l'analyse bivariée. On peut retrouver l'ensemble des graphiques observés dans le fichier `main.ipynb`.

4.2.1 Le genre du conducteur

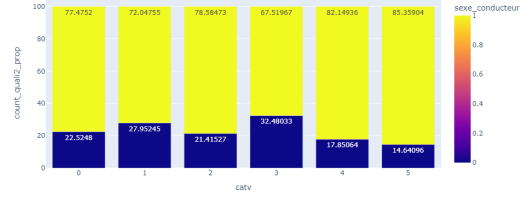
Un attribut qu'il est intéressant d'analyser est le genre du conducteur. En effet, il peut être source de biais s'il y a un déséquilibre entre hommes et femmes. On retrouve globalement la même proportion dans la corrélation que l'on soit homme ou femme. Les hommes étant beaucoup plus représentés dans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

La figure 8a nous montre une proportion de femmes 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 on 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 forcément un accident mortel, ou bien 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 L'âge du conducteur

On peut remarquer sur la figure 9 que la probabilité d'accident mortel beaucoup plus élevée à l'âge de 21 ans. Cela est dû en partie au fait que les conducteurs de 21 ans sont surreprésentés. Cette probabilité risque de poser problème pour notre modèle. En effet l'âge n'est pas un paramètre déterminant dans l'évaluation de la gravité de l'accident. Le problème est que notre modèle va probablement associer l'âge de 21 ans à un accident mortel, peu importe les autres paramètres de l'accident. L'âge serait donc notre **attribut sensible**.

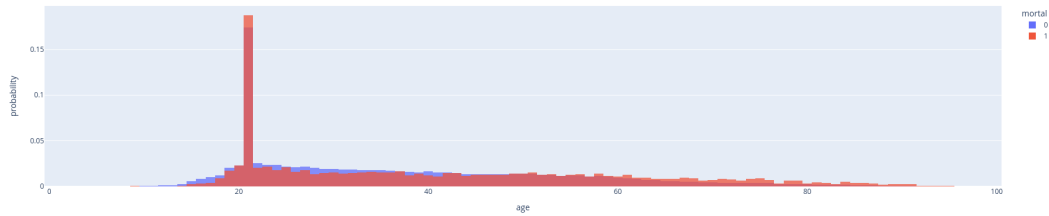


FIGURE 9 – Analyse bivariable âge – accident mortel

5 Apprentissage

5.1 Split

Séparer nos données en deux ensembles demandait de tenir compte d'une spécificité. Nous nous plaçons du point de vue d'un véhicule, mais bien souvent, plusieurs véhicules sont impliqués dans un même accident. Il nous a donc fallu adapter le split pour que les véhicules d'un même accident soient dans le même ensemble.

Par ailleurs, nous avons constaté que, si l'on ne le faisait pas, un surapprentissage lors du test se produisait. Typiquement, un trop grand nombre de morts étaient correctement prédits. Mais le modèle trouvait une partie de ces véhicules en les associant à ceux avec lesquels ils avaient eu un accident dans l'ensemble d'apprentissage.

5.2 Encodage One Hot

La base de données nous fournit pour la plupart des attributs des données qui peuvent être converties en entiers. Par exemple, une catégorie de véhicule (*catv*) est désignée par un chiffre.

L'attribut reste cependant catégoriel. En pratique, le seul attribut quantitatif qui nous reste est l'âge du conducteur. Le reste est soit binaire, soit qualitatif.

Tous les attributs qualitatifs ont donc été transformés en OneHot. Initialement, nous souhaitions optimiser le nombre de colonnes en réalisant les OneHot nous-mêmes. Par exemple, prenons "surf" (l'état de la surface de la route). Il est inutile de créer une colonne pour "Non-renseigné", "Autre" ou "normale". Nous sommes uniquement intéressés par les états spécifiques de la route.

Cependant, cette approche posait des problèmes car elle empêchait la réalisation de l'audit. Nous avons donc utilisé la méthode standard pour nos OneHot.

5.3 Arbre de décision

Une fois les données prêtes, nous avons utilisé le classifieur d'arbres aléatoires pour obtenir un modèle. C'est le premier que nous avons utilisé. En ce qui concerne les résultats, le score d'exactitude est de 1 pour l'entraînement et de 0.9014 pour le test.

En ce qui concerne la matrice de confusion, on remarque que 27687 véhicules ont été classés correctement en tant qu'accidents non létaux. Tandis que 291 accidents mortels ont été correctement identifiés. Cependant, 1413 accidents mortels ont été recensés à tort comme étant non létaux et 1648 véhicules ont subi le sort inverse.

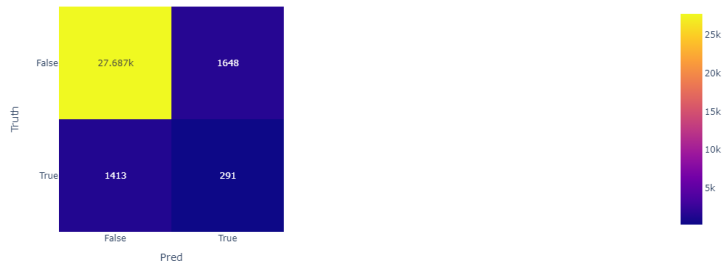


FIGURE 10 – La matrice de confusion pour le classifieur d'arbres aléatoires

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

5.4 GaussianNB

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

Il ne maximise pas notre score de précision puisqu'il n'est que de 0.8734 à l'entraînement 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 mortels recensés à tort comme étant non létaux à seulement 1196. Ceci entraîne malheureusement une perte de performances pour ce qui est de la détection d'accidents non mortels comme on peut le voir ci-dessous.



FIGURE 11 – La matrice de confusion avec GaussianNB

Un autre avantage qui peut ressembler à un inconvénient au départ est que ce classifieur demande des nombres et non des OneHot. Or nous en avons à foison. En dépit de ça, nos valeurs au départ sont en format numérique, rendant donc nos données compatibles avec cette approche. Ceci nous ouvre alors la possibilité de tester d’autres choses puisque le classifieur donne des probabilités et non des arbres. Au premier rang desquels *xplique*.

Par souci de concision cependant, nous prendrons le modèle issu du *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 résultats diffèrent en bien des points de ceux observés avec le *random tree classifier*, preuve qu’ils ont une approche bien différente.

5.5 Base Rate

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

- **Sexe du conducteur 1** (un homme)
- **Disparate Impact** : 1.21767750298587 1.6331199049428085
- **P-rule disparate Impact** : 0.8212355057458954 0.6123249107266375
- **Démographie Parité** : 0.011747835864606336 0.02383701239031672

Ce qui est flagrant ici, c’est qu’il existe une disparité dans la prédiction d’accidents mortels pour les hommes. Allant par ailleurs dans le sens de ce qu’indique Disparate Impact par la suite. Notons cependant que la disparité démographique existe mais est très faible.

- **Piéton 1** (un piéton est impliqué)
- **Disparate Impact** : 1.0538191011165416 1.0968113720110815
- **P-rule Disparate Impact** : 0.9489294689576994 0.9117337999207913
- **Démographie Parité** : 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 *disparate 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 variables 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 l'attribut *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 allons analyser 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 rôle prédominant de l'âge 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 y a 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 pratiquée 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 :

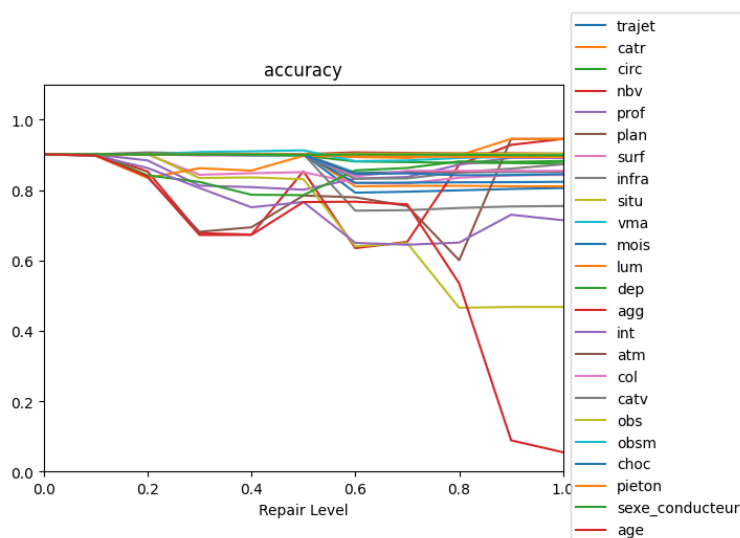


FIGURE 12 – L'accuracy en fonction du niveau de réparation pour chaque attribut.

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

On reconnaît les attributs dont l'accuracy est au plus haut en regardant ceux qui s'effondrent le plus. En premier lieu, l'âge s'illustre à nouveau. À la lumière de ceci, il apparaît évident que le modèle discrimine fortement les véhicules 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 paraît sensé, étant donné que certains lieux sont plus dangereux que d'autres. Enfin, notons que le type d'intersection et la catégorie du véhicule sont également sources de discriminations de la part de notre modèle. Encore une fois, cela semble cohérent.

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 collision, le type d'obstacle heurté ou encore le fait qu'un piéton soit impliqué. Pour ce qui est des piétons, cela peut paraître étonnant ; on peut émettre l'hypothèse que leur présence est souvent en ville là où les vitesses sont basses entraînant donc moins de morts.

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

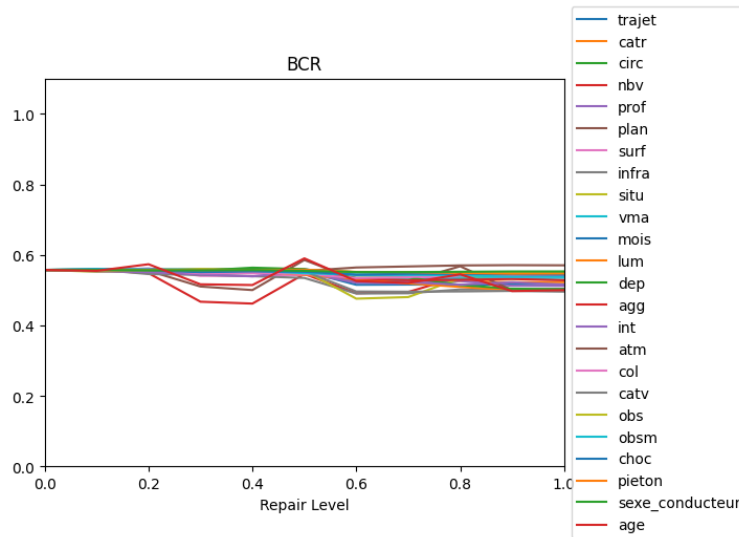


FIGURE 13 – Le BCR en fonction du niveau de réparation pour chaque attribut.

On remarque ici que les attributs sont plutôt centrés autour de 0.5 et que la plupart reste rectiligne, mais que certains 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 illustrés précédemment, à l'image du nombre de voies, du profil de la route ainsi que de la luminosité.

Malgré cela, l'évasement assez faible laisse penser que le modèle est plutôt équilibré. De plus, comme mentionné précédemment dans la section dédiée au split, nous avons pu tester le cas où on acceptait que des véhicules impliqués dans le même accident se retrouvent dans des ensembles différents (entraînement / test). Ce qu'il ressortait alors du BCR était un évasement beaucoup plus important et une origine située vers 0.68 . Ce qui laisse à penser 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, ce 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 une contribution plus importante 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.

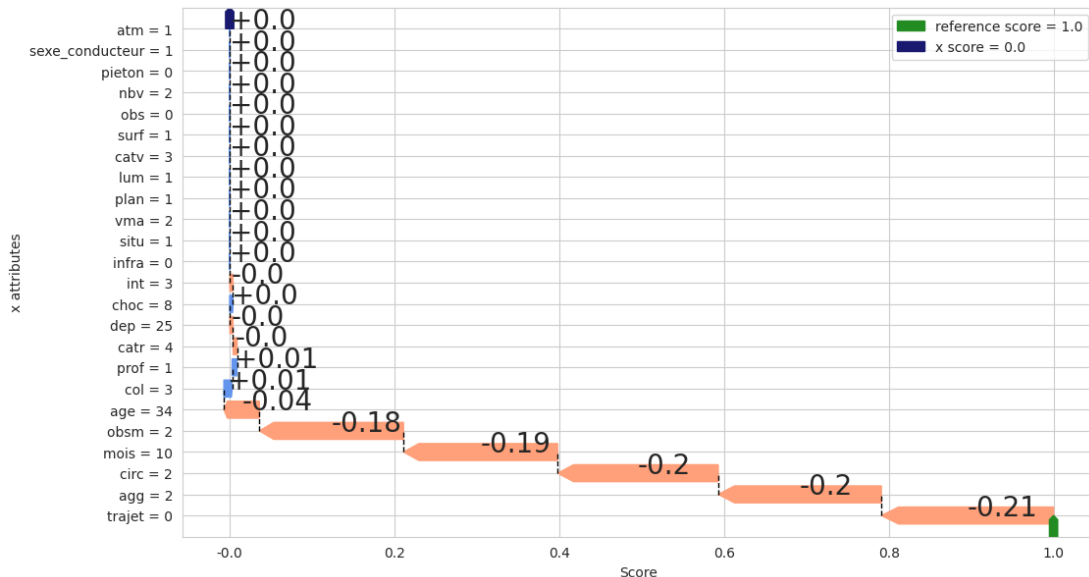


FIGURE 14 – Une approximation des valeurs de Shapley

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 avons pu l'observer avec les valeurs de Shapley, l'attribut *sexe_conducteur* ait un faible impact sur la prédiction.

7 Conclusion

Pour ce projet nous avons cherché un dataset qui nous intéressait, et nous avons choisi celui rapportant les accidents de la route. Pourvu de nombreux attributs, nous avons dû adapter leur format avant de pouvoir les analyser de manière univariée puis bivariée. Lors du premier essai, nous avons fait le choix d'utiliser des arbres de décision. Ce qui nous a conduit à transformer beaucoup de nos variables en OneHot. L'accuracy de ce modèle était élevée, mais il a du mal à trouver les accidents réellement mortels. Pour ce modèle, nous avons par la suite calculé le base rate puis effectué quelques audits tels que dice, BlackBoxAuditing et shapkit.

Par la suite nous avons essayé de trouver un classifieur qui avait de meilleures performances, c'est ainsi que nous avons trouvé GaussianNB, qui est meilleur pour détecter les véhicules impliqués dans des accidents vraiment mortels. Pour ce modèle nous avons aussi effectué les audits évoqués précédemment, mais nous ne les avons pas analysés dans ce rapport. Ce qui conclut notre premier rendu.

A Les données de notre dataset

Attribut	Description
<i>Num_Acc</i>	Numéro d'identifiant de l'accident
<i>jour mois</i>	Jour de l'accident, mois de l'accident
<i>an</i>	Année de l'accident
<i>hrmn</i>	Heure et minutes de l'accident
<i>lum</i>	Conditions d'éclairage dans lesquelles l'accident s'est produit
<i>dep</i>	Code INSEE du département
<i>com</i>	Numéro de commune (code INSEE)
<i>agg</i>	Localisation en agglomération
<i>int</i>	Type d'intersection
<i>atm</i>	Conditions atmosphériques
<i>col</i>	Type de collision
<i>adr</i>	Adresse postale (pour les accidents en agglomération)
<i>lat</i>	Latitude
<i>long</i>	Longitude
<i>catr</i>	Catégorie de route
<i>voie</i>	Numéro de la route
<i>circ</i>	Régime de circulation
<i>nbv</i>	Nombre total de voies de circulation
<i>vosp</i>	Présence d'une voie réservée
<i>prof</i>	Profil en long de la route
<i>plan</i>	Tracé en plan de la route
<i>surf</i>	État de la surface de la route
<i>infra</i>	Présence d'aménagements ou d'infrastructures
<i>situ</i>	Situation géographique de l'accident
<i>vma</i>	Vitesses maximale autorisées
<i>id_vehicule</i>	Identifiant du véhicule (clé étrangère)
<i>catv</i>	Catégorie du véhicule impliqué dans l'accident
<i>obs</i>	Type d'obstacle heurté
<i>obsm</i>	Type d'obstacle mobile heurté
<i>choc</i>	Point de choc initial
<i>manv</i>	Manœuvre principale avant l'accident
<i>catu</i>	Catégorie d'utilisateur (conducteur, passager, piéton)
<i>grav</i>	Gravité de l'accident pour l'utilisateur
<i>sexe</i>	Sexe du conducteur
<i>trajet</i>	Motif du déplacement au moment de l'accident
<i>mortal</i>	Indique si le véhicule est impliqué dans un accident mortel (calculé dans le notebook)