# Rapport technique 2

## Modèle & Performances des algorithmes

09.12.2022

Groupe 11

Nur Backiri
Miguel Guzman
Marianne Bellery
Emma Raulet

Ce rapport explique et interprète les résultats des implémentations réalisées dans le notebook commenté suivant :
https://colab.research.google.com/drive/19mLdkhQbsiJDrPKL4eGTDslXhgdp3UXr#scrollTo=7VzIGFMLwGAL

# Introduction

D'après l'Observatoire national interministériel de la sécurité routière, 2944 personnes ont perdu la vie dans un accident de la route en 2021. Malgré que ce chiffre soit en baisse par rapport à l'année de référence 2019 (hors Covid), le nombre d'accidents s'élève à plus de 50 000 par an. Afin d'améliorer la sécurité des automobilistes, motards, cyclistes et piétons, tous utilisateurs du réseau routier, on cherche à faire des préconisations d'aménagements routiers à mettre en place.

# Problématique métier

Grâce à notre analyse, nous allons permettre aux **instances gérant les routes** d'**améliorer le réseau routier** du point de vue de la **sécurité**

Nous allons déterminer les caractéristiques et différentes morphologies des routes sur lesquelles se produisent des accidents graves. Ainsi les instances gérant le réseau routier pourront les prendre en compte lors de créations, rénovation de routes ou encore pour modifier des portions de routes jugées trop dangereuses.

# Objectifs visés

Les objectifs que nous nous sommes visés sont les suivants :

- être capable d'identifier une zone accidentogène avant qu'un accident ait lieu.
- être capable de prédire la gravité de l'accident en fonction de la typologie de la zone concernée.

Ces deux premiers objectifs permettront aux responsables locaux d'être en mesure d'évaluer et de prédire le risque associé à la création ou l'extension d'un nouvel aménagement routier. Ainsi de justifier au vue des risques, sa mise en place ou bien d'adapter la signalétique, la prévention ou rajouter des aménagements de protection sur le réseau.

- étendre cette étude à une plus petite échelle, l'échelle départementale.

Ce dernier point permettra à l'échelle du département de mettre en place et de pré-tester ces mesures (quand elles modifient la circulation), ou bien de mettre en place des campagnes de sensibilisation en accord avec les risques ou les accidents ayant eu lieu au sein du département. En particulier, cela permettra aux responsables locaux comme les DIR, d'avoir une vue globale sur l'ensemble de leur zone d'action.

# Plan d'action

Afin de remplir les objectifs que nous nous sommes fixés, nous avons mis en place le plan d'action suivant :

1. Faire un clustering pour profiler les types de tronçons de routes à risque.

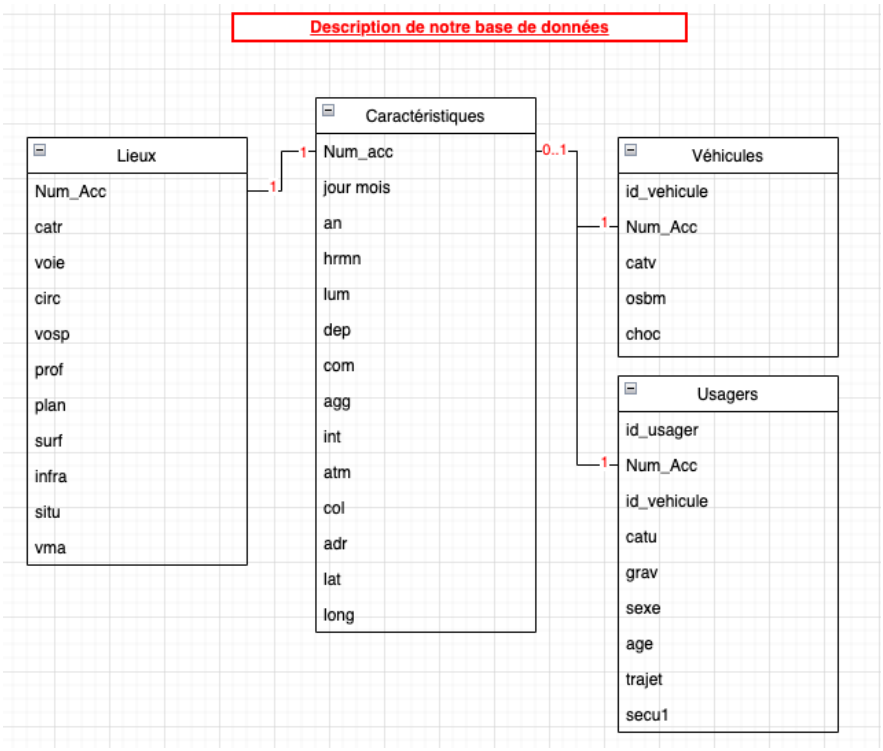
2. Entraîner des modèles de ML pour prédire la gravité d'un accident sous certaines conditions.
3. Entraîner un modèle de ML capable de prédire le nombre d'accidents/ de blessés par département sous certaines conditions.

# Table des matières

# Prétraitement des données

## 1. Jeux de données utilisés

Les données que nous avons utilisées durant ce projet sont extraites de la base de données annuelle des accidents corporels de la route pour l'année 2021, mise à disposition par le gouvernement via data.gouv. Cette base de données est constituée de 4 fichiers csv : Caractéristiques, Lieux, Véhicules et Usagers. Pour réaliser notre analyse nous avons combiné les 4 fichiers afin d'obtenir un unique dataset. Nous avons effectué nos jointures en utilisant Num_Acc : le numéro d'identification d'un accident.




Après avoir combiné nos différents datasets, on obtient un dataset dont une ligne représente un usager ou un véhicule inclus dans l'accident. Or on souhaite travailler sur un dataset dont une ligne représente un accident. Pour cela, on utilise la fonction **group by** sur la feature Num_Acc.

## 2. Ajout de nouvelles variables

Lors de la concaténation du dataset, on décide de créer de nouvelles variables, afin de ne pas perdre d'informations : en particulier, les informations sur les usagers impliqués dans l'accident.

On décide alors de créer les features suivantes, qui seront des variables explicatives :

| | | | |
|---|---|---|---|
| total_impliqués | nombre total de personnes impliquées dans l'accident/total | Int | Usagers |
| total_blessés | nombre total de personnes blessées dans l'accident | Int | Usagers |
| total_indemnes | nombre total de personnes indemnes dans l'accident | Int | Usagers |
| total_tués | nombre total de personnes tuées dans l'accident | Int | Usagers |
| total_blessés_graves | nombre total de personnes blessées graves dans l'accident | Int | Usagers |
| moy_age | Moyenne d'âge des blessés | Int | Usagers |

En parallèle, on décide de créer d'autres variables explicatives, à partir de la date de l'accident :

| | | | |
|---|---|---|---|
| saison | Saisonnalité | Int | Caractéristiques |
| date | date complète format 'dd/mm/yyyy' | Int | Caractéristiques |
| moment de la journée | Moment de la journée | Int | Caractéristiques |

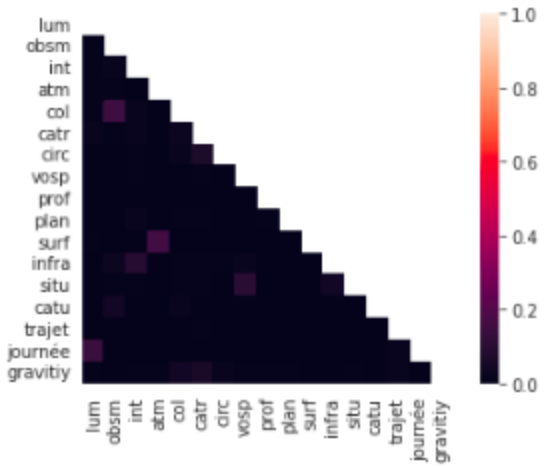
| | Num_Acc | Num_people | avg_age | total_hospitalisé | total_léger | total_Indemne | total_Tué | month | season | lum | dep | obsm | int | atm | col | lat | long | catr | circ | nbv | vosp | prof | plan | surf | infra |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 202100000001 | 2 | 32.0 | 1 | 0 | 1 | 0 | novembre | l'automne | Crépuscule ou aube | 30 | Véhicule | Hors intersection | Normale | Deux véhicules - frontale | 44.038958 | 4.348022 | Route Départementale | Bidirectionnelle | 2 | Sans objet | Plat | Partie rectiligne | Normale | Aucun |
| 1 | 202100000002 | 2 | 33.0 | 1 | 1 | 0 | 0 | septembre | l'automne | Plein jour | 51 | Véhicule | Intersection en T | Normale | Deux véhicules – par le coté | 49.242129 | 4.554546 | Route Départementale | Bidirectionnelle | 2 | Sans objet | Plat | Partie rectiligne | Normale | Aucun |
| 2 | 202100000003 | 2 | 44.0 | 1 | 0 | 1 | 0 | juillet | l'été | Plein jour | 85 | Piéton | Hors intersection | Temps éblouissant | Autre collision | 46.921950 | -0.964460 | Voie Communales | Bidirectionnelle | 2 | Sans objet | Plat | Partie rectiligne | Normale | Aucun |
| 3 | 202100000004 | 2 | 14.0 | 0 | 0 | 1 | 1 | mars | le printemps | Nuit avec éclairage public allumé | 93 | Véhicule | Intersection en X | Pluie forte | Autre collision | 48.949363 | 2.519664 | Route Départementale | Bidirectionnelle | 4 | Sans objet | Plat | Partie rectiligne | Normale | Aucun |
| 4 | 202100000005 | 1 | 24.0 | 0 | 1 | 0 | 0 | février | l'hiver | Nuit avec éclairage public allumé | 76 | Véhicule | Hors intersection | Normale | Deux véhicules – par l'arrière | 49.408380 | 1.145810 | Routes de métropole urbaine | Bidirectionnelle | 2 | Non renseigné | Plat | Partie rectiligne | Normale | Aucun |

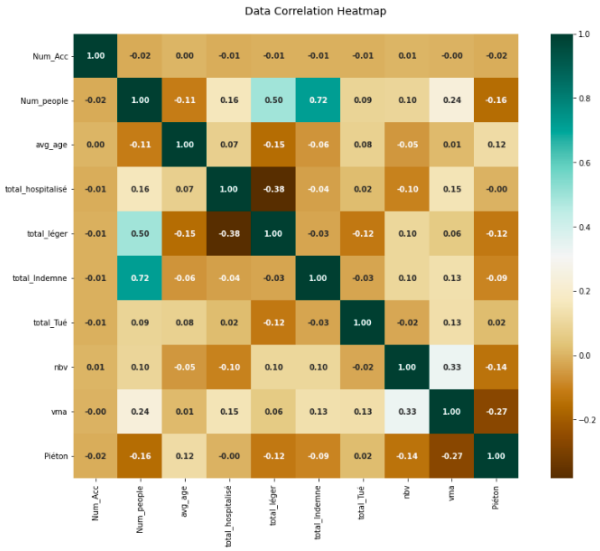
Le dataset obtenu contient l'ensemble des données que l'on utilisera pour appliquer nos algorithmes de machine learning. En voici, un extrait :

Par ailleurs, en fonction du type d'algorithme utilisé, on modifiera légèrement le dataset décrit précédemment.

## Analyse des corrélations pour simplifier le dataset utilisé

Afin de faire une première analyse des corrélations inhérentes au dataset que l'on va utiliser la suite du projet, on décide de réaliser deux heatmap. La première permet d'examiner les corrélations entre les variables catégoriques de notre dataset. Pour réaliser cette heatmap, on utilise la méthode dite de Cramer (à gauche). La seconde permet d'identifier les corrélations entre les variables numériques de notre dataset.




Les heatmap implémentées ne permettent pas d'identifier des corrélations fortes entre les différentes features, elles ne permettent pas de conclure. On ne peut donc pas simplifier facilement le dataset. Pour l'instant, on choisit de conserver l'ensemble des features sélectionnées précédemment.

# Profil des tronçons de routes accidentogènes

## 1. Choix des modèles utilisés

Le but de cette partie est de segmenter les données pour créer des classes de routes fortement accidentogènes. On parle de route mais en réalité, notre attention se porte sur des tronçons de routes ou bien des "zones" accidentogènes. On tente alors de classer les zones accidentogènes afin d'obtenir une liste de caractéristiques associées à chaque cluster.

Pour réaliser cette tâche, on choisit d'appliquer un algorithme d'apprentissage non supervisé, plus particulièrement un algorithme de classification automatique, plus communément appelé clustering.

Cependant, nous ne pouvons pas utiliser les algorithmes classiques de clustering comme par exemple k-means, puisqu'il est uniquement applicable sur des features numériques. Comme notre dataset est composé de features catégoriques, on choisit d'appliquer les algorithmes de clustering : Kmodes et Kprototypes.

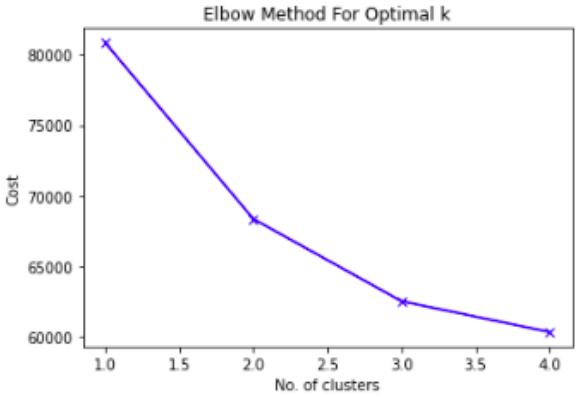
## 2. Modèle de clustering utilisés pour déterminer la typologie des routes

Puisque cette partie on veut créer une typologie des zones accidentogènes, on ne s'intéresse pas à toutes les données du dataset initial mais seulement aux features donnant une information précise sur la morphologie de la route à l'endroit de l'accident. A partir du dataset initial, on décide de supprimer toutes les features liées aux victimes (total_tués, …) et aux circonstances de l'accident (météo, moment de la journée, …). On conserve alors les features

suivantes : 'Num_Acc','int','catr', 'circ','vosp','prof', 'plan','infra', 'situ','surf'. Voici un extrait du dataset obtenu :

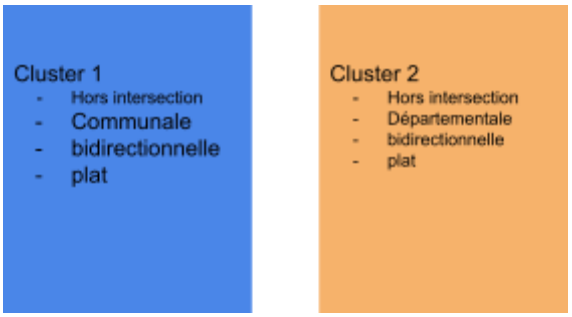
| Num_Acc | int | catr | circ | vosp | prof | plan | infra | situ | surf |
|---|---|---|---|---|---|---|---|---|---|
| 202100000001 | Hors intersection | Route Départementale | Bidirectionnelle | Sans objet | Plat | Partie rectiligne | Aucun | Sur chaussée | Normale |
| 202100000002 | Intersection en T | Route Départementale | Bidirectionnelle | Sans objet | Plat | Partie rectiligne | Aucun | Sur chaussée | Normale |
| 202100000003 | Hors intersection | Voie Communales | Bidirectionnelle | Sans objet | Plat | Partie rectiligne | Aucun | Sur chaussée | Normale |
| 202100000004 | Intersection en X | Route Départementale | Bidirectionnelle | Sans objet | Plat | Partie rectiligne | Aucun | Sur chaussée | Normale |
| 202100000005 | Hors intersection | Routes de métropole urbaine | Bidirectionnelle | Non renseigné | Plat | Partie rectiligne | Aucun | Sur chaussée | Normale |

Ensuite, afin de trouver le nombre optimal de cluster, on utilise la méthode Elbow. On obtient le résultat suivant :



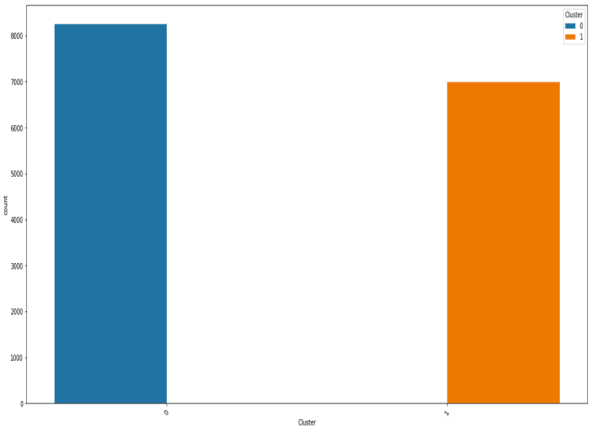

On choisit de prendre K = 2, donc 2 clusters.
La méthode de k-modes et de k-prototypes donnent des résultats comparables, on affichera les résultats

**Résultats obtenus pour K-modes (K=2) :**
Les clusters sont inégalement répartis et classent mal les zones accidentogènes puisqu'ils ont de nombreuses caractéristiques communes.









On remarque que les deux clusters regroupent des morphologies de routes quasiment identiques. Les résultats ne sont pas concluants. Afin d'améliorer les résultats du clustering, on suppose qu'il faut filtrer les données utilisées afin d'obtenir des classes de routes bien plus spécifiques et mieux triées.

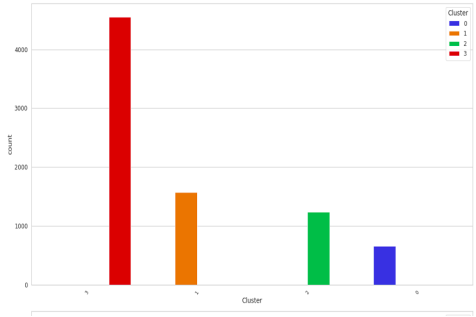
## 3. Amélioration des performances du clustering

Afin d'améliorer les résultats de notre clustering, on décide de rajouter la composante **gravity** dans notre dataset, et de ne filter nos données.

On décide d'effectuer un clustering après avoir largement simplifié notre dataset d'entrée:

- d'abord, on s'intéresse seulement aux zones accidentogènes dit 'serious', c'est-à-dire aux accidents sévères dont les victimes sont blessées.
- puis, on choisit seulement les accidents ayant eu lieu sur des 'voies communales' ou des 'routes de métropoles urbaines'. En effet, faire des clusters en associant un type d'intersection à une autoroute semble vain.

Les résultats sont les suivants :




Le clustering trie les données en créant des classes par types d'intersections. Cependant, on remarque que les résultats ne sont pas concluants puisque les autres caractéristiques ne sont pas propres à un cluster mais communes à tous. La segmentation en classe des types de routes n'est donc pas utilisable.

## 4. Conclusion

On remarque que les clustering effectués permettent de créer des profils de routes assez limités. En effet, les caractéristiques sont souvent communes donc il est difficile de tirer des conclusions.

On remarque cependant que les zones plus accidentogènes sont les voies communales et les départementales.

Dans cette partie, l'objectif était de segmenter les données afin de créer un vue des différentes morphologies de routes accidentogènes. Le but visé est d'être en capacité de repérer les zones accidentogènes lors de la construction ou l'extension du réseau routier afin d'être en mesure d'optimiser la sécurité associée à celui-ci ou bien de mettre en place une signalétique ou politique de sensibilisation adaptée.

# Modèles de classification utilisés pour la prédiction de la gravité

## 1. Préparation du dataset

Pour appliquer nos modèles de prédiction de la gravité, il nous faut modifier notre dataset : on supprime les colonnes à partir desquelles on a calculé la gravité (feature gravity). On obtient le dataset dont voici un extrait :

| | Num_people | avg_age | lum | obsm | int | atm | col | catr | circ | nbv | vosp | prof | plan | surf | infra | situ | vma | catu | trajet | gravitiy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 32.0 | Crépuscule ou aube | Véhicule | Hors intersection | Normale | Deux véhicules - frontale | Route Départementale | Bidirectionnelle | 2 | Sans objet | Plat | Partie rectiligne | Normale | Aucun | Sur chaussée | 80 | Conducteur | Domicile – travail | serious |
| 1 | 2 | 33.0 | Plein jour | Véhicule | Intersection en T | Normale | Deux véhicules – par le coté | Route Départementale | Bidirectionnelle | 2 | Sans objet | Plat | Partie rectiligne | Normale | Aucun | Sur chaussée | 80 | Conducteur | Non renseigné | serious |
| 2 | 2 | 44.0 | Plein jour | Piéton | Hors intersection | Temps éblouissant | Autre collision | Voie Communales | Bidirectionnelle | 2 | Sans objet | Plat | Partie rectiligne | Normale | Aucun | Sur chaussée | 50 | Conducteur | Domicile – travail | serious |
| 3 | 2 | 14.0 | Nuit avec éclairage public allumé | Véhicule | Intersection en X | Pluie forte | Autre collision | Route Départementale | Bidirectionnelle | 4 | Sans objet | Plat | Partie rectiligne | Normale | Aucun | Sur chaussée | 50 | Conducteur | Non renseigné | serious |
| 4 | 1 | 24.0 | Nuit avec éclairage public allumé | Véhicule | Hors intersection | Normale | Deux véhicules – par l'arrière | Routes de métropole urbaine | Bidirectionnelle | 2 | Non renseigné | Plat | Partie rectiligne | Normale | Aucun | Sur chaussée | 50 | Conducteur | Domicile – travail | not serious |

Pour rendre ce dataset utilisable, on encode les différentes features. Rapidement, voici la listes des étapes de traitement de ce dataset :

- on encode toutes les features categorical en utilisant get.dummies. Ce qui augmente largement le nombre de features dans notre dataset.
- on encode la feature gravity grâce à LabelEncoder extrait de sklearn preprocessing. On obtient {0,1} avec 0=not serious, 1=serious.
- enfin on normalise les features numériques en utilisant MinMaxSaler form sknlearn preprocessing.

Ensuite, on découpe notre dataset en plusieurs sous ensemble de données selon un rapport 0.65= train /0.35 = test.

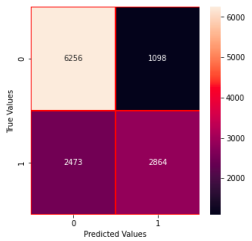
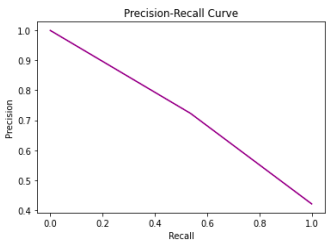
Ce prétraitement additionnel des données nous permet de préparer notre dataset à application des différents algorithmes de ML que nous allons utiliser dans la suite.

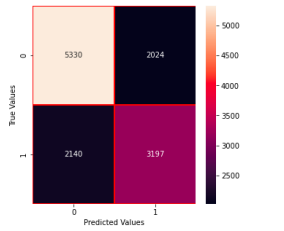
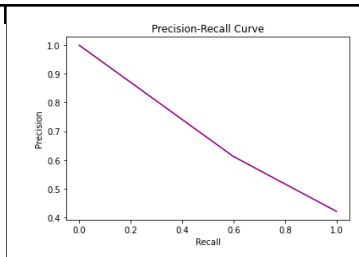
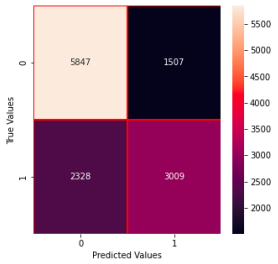
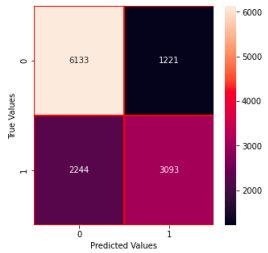
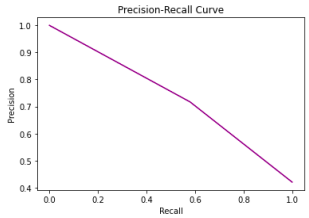
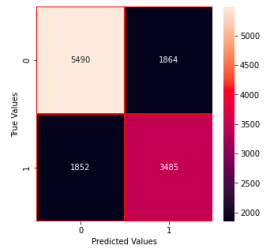
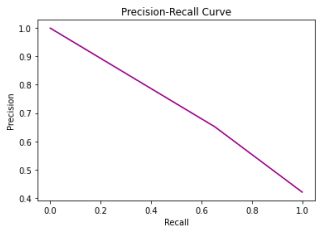
## 2. Choix des modèles de classification utilisés et justification des paramètres choisis

| Modèle | Choix du modèle | Hyperparamètres du modèle |
|---|---|---|
| Régression logistique | Nous avons utilisé un modèle de régression logistique avec pénalité élastique (lasso, crête) et l1 (lasso). Ensuite, nous avons utilisé le changement de poids de classe car notre distribution cible n'est pas équilibrée. Après nous avons vu que notre modèle s'améliorait en vérifiant le résultat f1. | penalty =elasticnet,l1<br>solver = saga,liblinear<br>l1_ratio=0.6<br>max_iter = 2000 |
| SVM | SVM avec des noyaux non linéaires peut transformer des | kernel = 'rbf' (Gaussian |

| | | |
|---|---|---|
| | fonctionnalités existantes en fonctionnalités de grande dimension qui peuvent être séparées linéairement dans des espaces de plus grande dimension. Puisque nous avons une cible déséquilibrée, il est bon d'essayer SVM. | Radial Basis Function)<br>class_weight="balanced" |
| Random Forest | Le Random Forest est une combinaison d'arbres tels que chaque arbre dépend d'un sous-ensemble aléatoire d'entités et de données. En conséquence, chaque arbre de la forêt est différent et fonctionne généralement mieux que l'ensachage. Les paramètres les plus importants sont le nombre d'arbres et le nombre d'entités à échantillonner. L'augmentation du nombre d'arbres améliore les résultats et n'entraîne pas de surajustement dans la plupart des cas, mais les améliorations plafonnent à mesure que vous ajoutez d'autres arbres. De plus, nous avons déterminé le max. fonctionnalités en utilisant la racine carrée de nos fonctionnalités -1. Et nous avons vu que la forêt aléatoire fonctionne mieux que SVM. | n_estimators=50<br>max_features=9 |
| GXBoost | XGBoost est un type de modèle d'amplification de gradient qui peut être utilisé pour des tâches de classification. Il est souvent utilisé avec des données tabulaires, qui incluent des données catégorielles. XGBoost est connu pour son efficience et son efficacité. | verbosity=2<br>max_depth=5<br>objective='binary:logistic' |
| Decision tree | Decision Trees peuvent gérer les données catégorielles de manière native, sans avoir besoin d'un encodage spécial. De plus, les arbres de décision sont faciles à interpréter et à comprendre, ce qui peut être utile lorsque vous travaillez avec des ensembles de données complexes. Cependant, les arbres de décision peuvent également être sujets à un surajustement, en particulier lorsque vous travaillez avec des ensembles de données volumineux et complexes. La question de savoir si les arbres de décision constituent le meilleur choix pour un ensemble de données catégorielles spécifique dépendra de nombreux facteurs, tels que la taille et la qualité de l'ensemble de données et les objectifs spécifiques du modèle. Afin d'éviter le surajustement, nous avons essayé d'utiliser gridsearchcv pour trouver les meilleurs paramètres et nous avons également essayé d'utiliser un modèle de forêt aléatoire. | We used gridsearchcv in order to find best hyperparameters for decision tree model.<br>params_grid = {<br>'criterion': ['gini', 'entropy'],<br>'max_depth': [5, 10, 15, 20],<br>'min_samples_leaf': [1, 2, 5]<br>}<br>And gridsearchcv gave us best parameters: {'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 5} |

## 3. Modèles utilisés et évaluation de leurs performances

| Modèle | Métriques | Matrice de confusion | Courbe Roc | Commentaires |
|---|---|---|---|---|
| Régression logistique | Accuracy is: 0.72<br>Precision is: 0.72<br>Recall is: 0.54<br>Fscore is: 0.54<br>AUC is: 0.69 |  |  | On peut classer les modèles utilisés selon leurs performances :<br>- Les modèles de Decision tree et de Random Forest obtiennent les meilleurs |

| | | | | |
|---|---|---|---|---|
| SVM | Accuracy is: 0.67<br>Precision is: 0.61<br>Recall is: 0.60<br>Fscore is: 0.60<br>AUC is: 0.66 |  |  | résultats. L'accuracy et le F1Score sont > 0.60.<br>- Le GXBoost aussi obtient de bons résultats même si son Accuracy est moins bonne.<br>- Cependant, on remarque que les modèles de régression logistique et de SVM obtiennent de mauvais résultats.<br>On cherchera dans la suite à améliorer les performances de certains modèles. |
| Random Forest | Test Accuracy = 0.697<br>Train Accuracy = 0.996 |  | | |
| GXBoost | Accuracy is: 0.73<br>Precision is: 0.72<br>Recall is: 0.58<br>Fscore is: 0.58<br>AUC is: 0.71 |  |  | |
| Decision tree | Accuracy is: 0.71<br>Precision is: 0.65<br>Recall is: 0.65<br>Fscore is: 0.65<br>AUC is: 0.70 |  |  | |

## 4. Amélioration des modèles utilisés

*4.1 ) Pondération du data set*

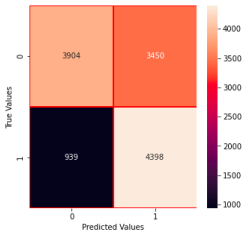
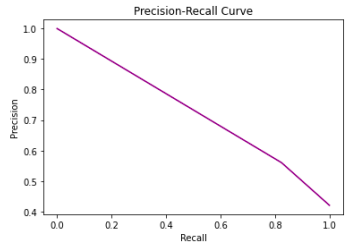
Lorsqu'on évalue les performances de la régression logistique, on remarque que le F1 Score est seulement de 0.54. On remarque en parallèle que le dataset utilisé en entrée est inégalement distribué : 15000 accidents sont considérés 'serious' (gravity=1) et 21000 sont considérés 'not serious'.

Afin d'améliorer les résultats du modèle, on décide de rajouter des pondérations. Les poids attribués à chacune des classes sont choisis de telle sorte à effacer la disparité entre les accidents 'serious' et 'not serious'.

Les pondérations ont été implémentées de la manière suivante :

```
class_weight = {}
# Assign weight of class 0 to be 0.3
class_weight[0] = 0.3
# Assign weight of class 1 to be 0.7
class_weight[1] = 0.7
```

En rééquilibrant le dataset grâce aux pondérations, on obtient de bien meilleurs résultats :

| | | | |
|---|---|---|---|
| Régression logistique V3 | Accuracy is: 0.65<br>Precision is: 0.56<br>Recall is: 0.82<br>Fscore is: 0.81<br>AUC is: 0.68 |  |  |

## 5. Conclusion

Sans modifier le dataset, les modèles que nous avons utilisés ont de bonnes performances (en particulier Decision tree et Random Forest). L'utilisation d'un Random Forest peut être privilégiée car même si il est plus complexe, il obtient de meilleures performances que le décision tree (en témoigne le Test_accuracy > 0.99).

Dans cette partie, l'objectif était de prédire à l'échelle d'un accident la dangerosité d'une route ou d'un tronçon de route. En effet, le but est de permettre aux responsables locaux de la maintenance et de l'entretien des routes, d'être capable d'évaluer la dangerosité des aménagements routiers avant qu'ils soient ouverts à la circulation et donc d'optimiser la signalétique et les éléments de sécurité associés.

Notre algorithme de régression logistique, bien qu'il nécessite de rajouter une pondération sur les données d'entrée, obtient de bons résultats, et pourra aussi être utilisé pour produire l'effet escompté.

A partir des conditions de circulation initiales, en particulier sur la typologie des routes (intersection, pente, …), l'algorithme est capable de prédire la gravité d'un accident. L'intérêt pour les responsables locaux réside en la capacité de l'algorithme à prendre en compte des données météorologiques : luminosité, moment de la journée, météo, … Ce qui peut être pratique pour ajuster en fonction des différents départements. Par exemple, à Brest, le risque de pluie étant très fréquent, on peut penser que la collectivité en charge des routes pourrait faire des prédictions adaptées au climat.

# Prédiction du nombre de morts pour chaque département

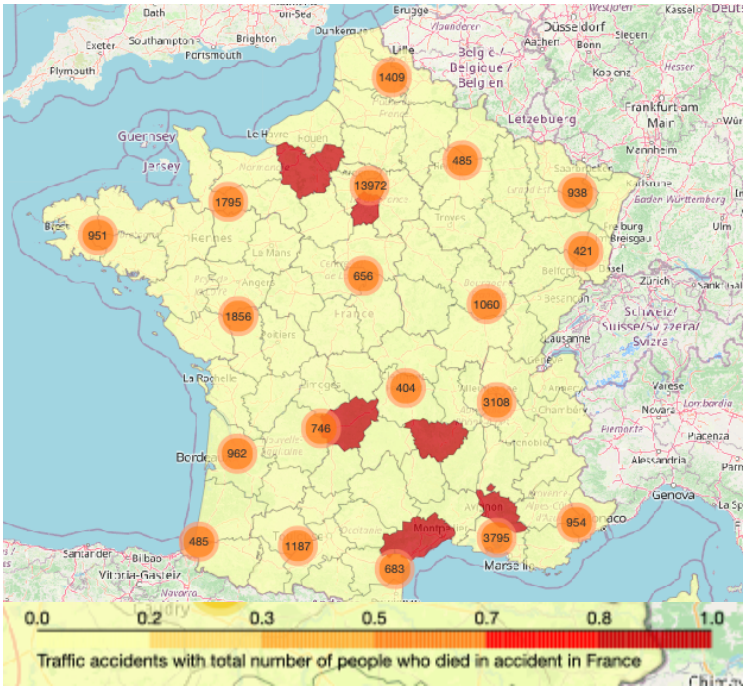
## 1. Quels sont les départements les plus meurtriers ?

Lors de l'exploration de nos données, nous avons décidé de créer une carte interactive permettant de visualiser le nombre de morts par département.

Les départements les plus meurtriers sont :

- Eure
- Essonne
- Hérault
- Vaucluse
- Corrèze
- Haute Loire

Cette analyse nous permet d'obtenir un premier aperçu des départements les plus concernés par nos futures recommandations.



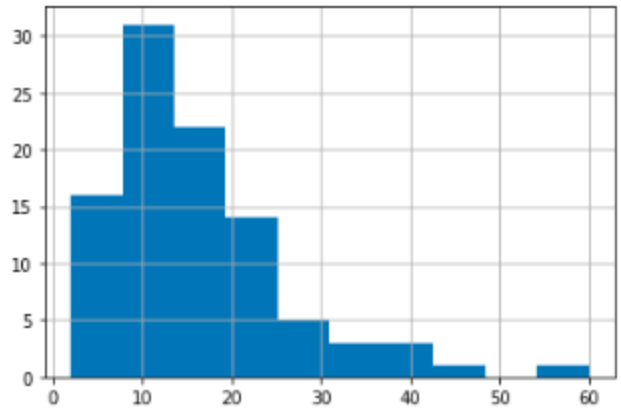

## 2. Modèles de régression pour la prédiction du nombre de morts

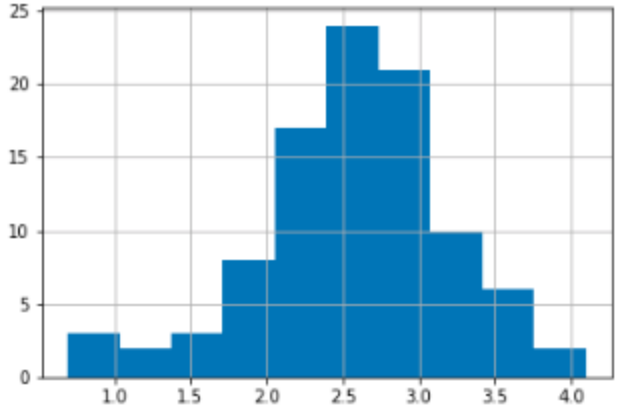
Afin d'affiner l'étude des accidents mortels sur les routes, on décide d'implémenter un modèle de régression linéaire permettant de prédire le nombre de morts par département.

A partir de notre dataset initial, on effectue une opération de **group by** sur les départements (**dep**). On décide alors de sommer les informations associées aux accidents liés à chaque département. En voici un extrait :

| dep | Num_people | total_Tué | lum_Crépuscule ou aube | lum_Nuit avec éclairage public allumé | lum_Nuit avec éclairage public non allumé | lum_Nuit sans éclairage public | lum_Plein jour | obsm_Animal domestique | obsm_Animal sauvage | obsm_Non renseigné | obsm_Piéto |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 01 | 658 | 16.0 | 15.0 | 16.0 | 2.0 | 31.0 | 171.0 | 0.0 | 0.0 | 0.0 | 21. |
| 02 | 259 | 22.0 | 5.0 | 6.0 | 0.0 | 16.0 | 75.0 | 0.0 | 1.0 | 0.0 | 18. |
| 03 | 351 | 7.0 | 10.0 | 11.0 | 1.0 | 14.0 | 97.0 | 1.0 | 2.0 | 0.0 | 22. |
| 04 | 285 | 7.0 | 6.0 | 3.0 | 0.0 | 7.0 | 86.0 | 1.0 | 1.0 | 0.0 | 11. |
| 05 | 444 | 12.0 | 1.0 | 10.0 | 1.0 | 6.0 | 146.0 | 1.0 | 0.0 | 0.0 | 20. |

Afin de prédire le nombre de morts, on choisit d'abord un algorithme de régression linéaire. Les résultats n'étant pas cohérents, on décide de modifier le type de régression. En visualisant la distribution des valeurs d'entrée de l'algorithme, on décide d'appliquer une fonction log qui permet de recentrer la distribution de la valeur à prédire ici : **total_tués**.




Le r2 score est évalué à 0.48 et l'erreur quadratique en moyenne à 0.4.

### 3. Conclusion

Les résultats de la régression ne sont pas véritablement concluants, en effet on voit bien que les métriques choisies indiquent de faible taux de réussite.

Afin d'améliorer ce modèle, il faudrait simplifier les données d'entrée de la régression par exemple en réduisant le nombre de features.

## Conclusion

L'étude que nous avons est concluante, nous avons bien réussi à compléter les objectifs énoncés au début du projet, qui était d'identifier des zones accidentogènes et prédire la sévérité des accidents à plusieurs échelles.

Même si certains de nos modèles ne sont pas tout à fait performants, ils restent utilisables.

Afin de compléter et affiner notre étude, on pourrait penser aux perspectives d'amélioration suivantes :

- pour améliorer les clustering, il faudrait simplifier encore le dataset utilisé pour n'obtenir que certaines features (les plus pertinentes) et éventuellement filtrer pour n'obtenir qu'une partie des lignes, en supprimant les features utilisées sur ces filtres.
- pour améliorer nos modèles de prédiction de la gravité des accidents, on peut penser à utiliser un dataset pondéré pour chaque modèle. On peut aussi complexifier la formule de la gravité (ici 'serious' ou 'not serious') en choisissant une modélisation mathématique complexe de la gravité (prenant par exemple en compte le nombre de passagers, le nombre de morts, …).

- pour améliorer la régression, il faudrait de la même manière que pour le clustering, utiliser un dataset simplifier en supprimant les features les moins pertinentes.

Enfin de manière plus globale, on pourrait penser aux axes suivants :

- ajouter une dimension temporelle en combinant chaque base de données annuelle en une unique base de données qui s'étendait de 2010 à 2021 (par exemple)

# Annexes

- Analyse exploratoire des données :

En parallèle du travail précédent que nous avons réalisé, nous avons travaillé à l'analyse des données, celle-ci permet aussi de tirer des conclusions et de compléter le projet mené. C'est pourquoi nous avons jugé utile d'inclure certains axes de réflexions dans ce rapport.

- Figure 1 : Correlation heatmap


- Figure 3 : Tableau résumant les features du dataset final utlisé

| Nom | Description | Type | Utilisation | Provenance |
|---|---|---|---|---|
| Informations générales sur les accidents | | | | |
| Num_Acc | Numéros d'identifiant de l'accident | Int | permet de faire la jointure entre les différents datasets | Caractéristiques |
| lat | Latitude | Float | permet de cartographier | Lieux |
| long | Longitude | Float | permet de cartographier | Lieux |
| Circonstances de l'accident | | | | |
| saison | Saisonnalité | Int | donne la saison | Caractéristiques |
| date | date complète format dd/mm/yyyy | Int | donne la date | Caractéristiques |
| moment de la journée | Moment de la journée | Int | | Caractéristiques |
| lum | Lumière/éclairage | Int | variables explicatives donnant la qualité de l'éclairage lors de l'accident | Caractéristiques |

| | | | | |
|---|---|---|---|---|
| atm | Conditions atmosphériques | Int | variables explicatives donnant les conditions atmosphériques | Caractéristiques |
| surf | Surface de la route au moment de l'accident | Int | conditions | Lieux |
| Typologie des routes | | | | |
| int | Type d'intersection | Int | variables explicatives donnant la typologie de la route | caractéristiques |
| catr | Catégorie de la route | Int | variables explicatives donnant la typologie de la route | Lieux |
| circ | Circulation | Int | variables explicatives donnant la typologie de la route | Lieux |
| nbv | Nombre véhicules | Int | variables explicatives donnant la typologie de la route | Lieux |
| vosp | | | variables explicatives donnant la typologie de la route | Lieux |
| prof | | | variables explicatives donnant la typologie de la route | Lieux |
| plan | Typologie du virage | | variables explicatives donnant la typologie de la route | Lieux |
| infra | Infrastructure | | variables explicatives donnant la typologie de la route | Lieux |
| situ | | | variables explicatives donnant la typologie de la route | Lieux |
| vma | Vitesse maximale autorisée | | variables explicatives donnant la typologie de la route | Lieux |
| Dommages matériels | | | | |
| col | Type de collision | Int | donne la localisation de | Caractéristiques |
| obsm | Type d'obstacle mobile percuté | Int | piéton, animal, véhicule ? | Véhicules |
| is_pieton | obsm == 2 | Boolean | | Véhicules |
| grav_phy | | Int | | Véhicules |
| Dommages corporels | | | | |
| total_impliqués | nombre total de personnes impliquées dans l'accident/total | Int | | Usagers |
| total_blessés | nombre total de personnes blessées dans l'accident | Int | | Usagers |
| total_indemnes | nombre total de personnes indemnes dans l'accident | Int | | Usagers |
| total_tués | nombre total de personnes tuées dans l'accident | Int | | Usagers |

| | | | | |
|---|---|---|---|---|
| total_blessés_graves | nombre total de personnes blessées graves dans l'accident | Int | | Usagers |
| grav_corp | Gravité corporelle de l'accident???? | Int? | | Usagers |
| age_conducteur | Age du conducteur | Int | | Usagers |
| moy_age | Moyenne d'âge des blessés | Int | | Usagers |