Analyse et prédiction des accidents de la route

Wilfried BEMELINGUE
Pierre LAVIELLE
Tanguy DUCROCQ
Grégoire SUISSA

Table des matières

1 Introduction **4**

2 Prétraitement et nettoyage des données **4**

3 Prédiction de la sévérité des accidents **5**
3.1 Objectif 5
3.2 Étapes de traitement et modélisation 5
3.3 Meilleurs modèles 7
3.3.1 Équilibrage des données 7
3.4 Méthodologie 8
3.4.1 Algorithmes testés 8
3.4.2 Métriques d'évaluation 8
3.4.3 Optimisation des hyperparamètres 8
3.5 Résultats 9
3.5.1 Comparaison initiale des modèles 9
3.5.2 Optimisation des hyperparamètres 10
3.5.3 Performance du meilleur modèle 12
3.5.4 Analyse des résultats 13
3.5.5 Conclusion des modèles 13
3.5.6 Limites et améliorations 13

4 Segmentation (clustering) des accidents par profil **13**
4.1 Jeu de données 14
4.1.1 Questions 14
4.1.2 Ajout d'attributs 14
4.2 Méthode 15
4.2.1 Questions 15
4.2.2 Ajout d'attributs 15
4.3 Analyses 15
4.3.1 Questions 16
4.3.2 Ajouts d'attributs 22
4.4 conclusion 27

5 Détection automatique d'accidents anormaux à partir de données météorologiques, temporelles et spatiales **28**
5.1 Objectif du projet 28
5.2 Traitement et échantillonnage des données 28
5.3 Choix des algorithmes et justification 28
5.4 Exploration visuelle et analyse contextuelle 29
5.5 Prétraitement et encodage 29
5.6 Conclusion intermédiaire 31

5.7 Détection d'anomalies : méthodes appliquées et résultats observés 31
5.8 Réduction de dimension et visualisation des résultats 34
5.9 Analyse croisée et complémentarité des approches 36
5.10 Conclusion de chapitre 36

6 Conclusion **37**

1 Introduction

Ce rapport détaille l'Analyse et prédiction des accidents de la route du jeu de données US_Accidents_March23. L'objectif principal de ce projet est de traiter le dataset à travers plusieurs problématiques :

— Prétraitement et nettoyage des données
— Prédiction de la sévérité des accidents
— Segmentation (clustering) des accidents par profil
— Détection d'anomalies dans les accidents

2 Prétraitement et nettoyage des données

Avant de procéder à l'analyse exploratoire et à la modélisation, un nettoyage approfondi du jeu de données a été réalisé à l'aide de la bibliothèque PySpark. Cette phase de prétraitement a permis d'obtenir un jeu de données cohérent, sans valeurs manquantes, et structuré pour l'apprentissage automatique.

Dans un premier temps, le fichier `US_Accidents_March23.csv` a été chargé en utilisant la fonction `spark.read.csv` avec les options `header=True` et `inferSchema=True` pour détecter automatiquement les types de données et conserver les noms de colonnes.

Certaines colonnes ont été supprimées car elles ne présentaient pas de valeur ajoutée pour l'analyse ou étaient trop spécifiques. Parmi elles, on trouve des identifiants uniques, des champs descriptifs textuels peu exploitables en l'état, ainsi que des données géographiques redondantes ou incomplètes : `ID`, `Source`, `Description`, `Street`, `Zipcode`, `Airport_Code`, `End_Lat`, `End_Lng`, `Country`, `Timezone`, et `Weather_Timestamp`.

Les colonnes de type booléen ont été converties en valeurs entières (`0` ou `1`) afin de faciliter leur intégration dans les modèles de machine learning qui ne traitent pas directement les types booléens.

Une analyse des valeurs manquantes a ensuite été effectuée. Pour chaque colonne, le taux de données manquantes a été calculé en distinguant les colonnes numériques, où les `NaN` sont pris en compte, des colonnes non numériques, où seules les valeurs nulles sont comptabilisées.

Les valeurs manquantes ont été traitées par imputation selon le type de variable. Pour les colonnes numériques, la médiane a été utilisée comme valeur de remplacement, estimée avec la méthode `approxQuantile` pour conserver la robustesse face aux valeurs extrêmes. Pour les colonnes catégorielles, les valeurs manquantes ont été remplacées par la chaîne `"Unknown"` afin de préserver l'information tout en évitant la suppression d'observations.

À l'issue de ce prétraitement, le jeu de données ne contient plus de valeurs manquantes, et toutes les variables sont dans un format compatible avec les techniques d'analyse et de modélisation supervisée.

Néanmoins notre jeu de donnees possede des valeurs qui ne nous permettent pas d'utiliser les modeles , les `string` venant des attributs comme `Sunrise_Sunset`. Les valeurs de type `string` ont ete indexe transformant ces donnees en `integers`, cette transformations s'effectue pour chacun des objectifs permettant de faire un dictionnaire reliant les indexes et leur valeurs.

3 Prédiction de la sévérité des accidents

3.1 Objectif

L'objectif est de prédire la sévérité d'un accident (valeurs de 1 à 4) à partir de variables environnementales et contextuelles. Pour cela, un pipeline complet de préparation, modélisation et évaluation a été mis en place.

3.2 Étapes de traitement et modélisation

1. Sélection des variables Une sélection manuelle a été réalisée sur la base de l'exploration des données et de la logique métier. Les variables jugées peu informatives ou trop corrélées ont été supprimées pour éviter la redondance (exemple : suppression de X car trop corrélé à Y).

2. Création de nouvelles variables (Feature Engineering) Plusieurs variables ont été dérivées afin d'enrichir l'information initiale du dataset et d'améliorer la capacité prédictive des modèles :

- **Is_Weekend** : une variable binaire indiquant si l'accident a eu lieu durant un week-end. Cette information est obtenue à partir du jour de la semaine (`Weekday`), en considérant que les jours 1 (dimanche) et 7 (samedi) correspondent au week-end.
- **Heure, jour et mois de l'accident** : à partir du champ `Start_Time`, les colonnes `Hour`, `Weekday` et `Month` ont été extraites. Ces variables permettent de capturer des effets temporels potentiels, comme des pics d'accidents à certaines heures ou saisons.
- **Durée de l'accident** : une nouvelle variable `Accident_Duration`, exprimée en minutes, a été calculée en prenant la différence entre `End_Time` et `Start_Time`. Elle peut révéler la gravité ou la complexité de gestion de l'accident.

3. Prétraitement des données Un pipeline Spark ML a été construit avec les étapes suivantes :

- **Indexation des variables catégorielles** via `StringIndexer`
- **Encodage one-hot** avec `OneHotEncoder`
- **Assemblage des features** avec `VectorAssembler`
- **Standardisation** via `StandardScaler`

Problème rencontré : initialement, les étapes d'encodage avaient été faites hors pipeline, ce qui empêchait l'interprétation des coefficients du modèle final. **Solution :** l'encodage a été

intégré au pipeline complet pour conserver la traçabilité des transformations.

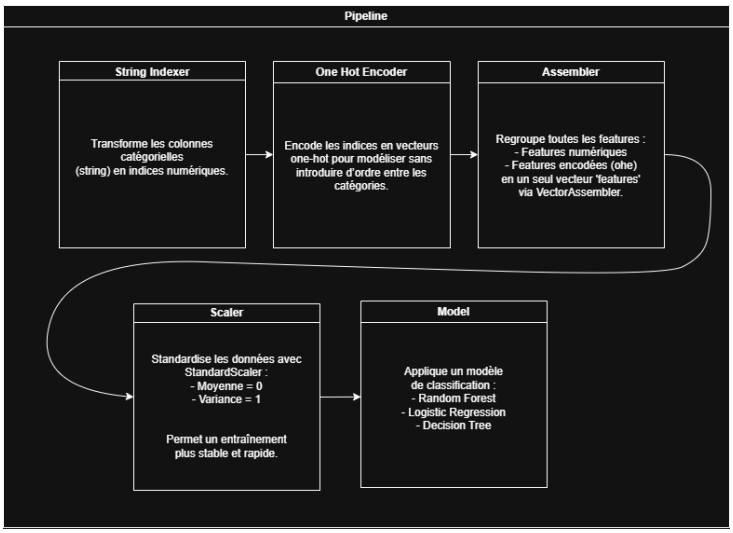


FIGURE 1 – Pipeline complet de traitement et d'entraînement

4. Premiers modèles Plusieurs modèles de classification ont été testés sur un sous-échantillon des données pour accélérer les premiers essais :

- `RandomForestClassifier`
- `LogisticRegression`
- `DecisionTreeClassifier`

Les résultats ont été évalués via les métriques F1 score, précision et rappel.

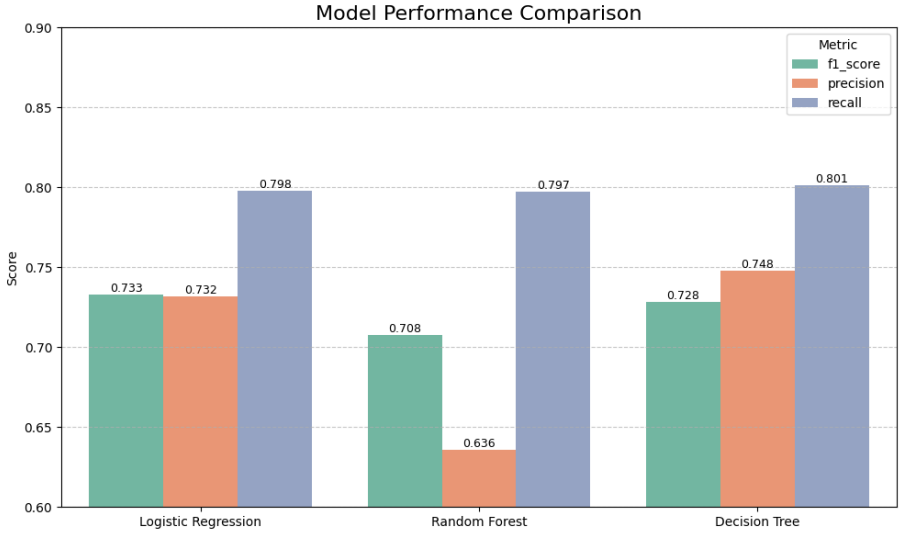


FIGURE 2 – Comparaison des différents model étudie

Problème identifié : bien que certains modèles aient un F1 score global élevé (~0.7), la prédiction était biaisée vers la classe 2, majoritaire à 80 %. **Solution :** analyse de la *matrice de confusion* pour révéler ce biais.

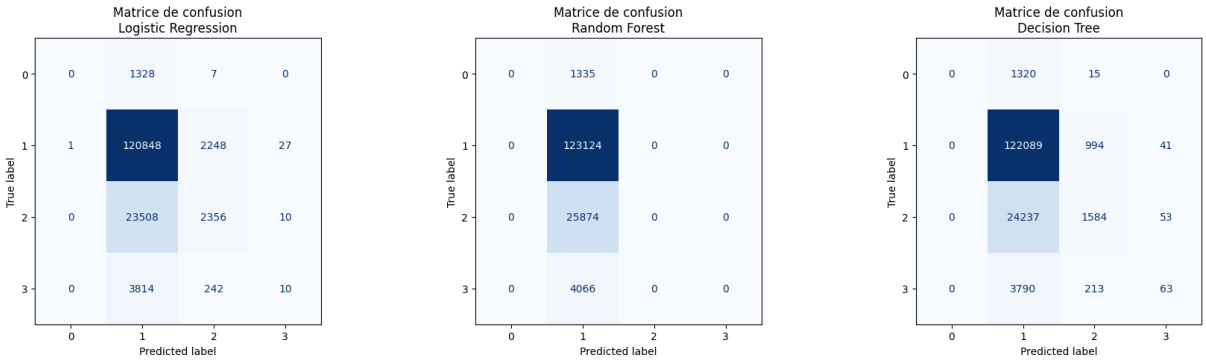


FIGURE 3 – Matrices de confusion des différents modèles

5. Gestion du déséquilibre des classes Le jeu de données présentait un fort déséquilibre entre les classes, rendant la classification difficile.

Severity	Train	Test	Train_Sample	Test_Sample	Train_%	Test_%	Train_Sample_%	Test_Sample_%
1	53955	13411	5259	1335	0.87	0.87	0.86	0.86
3	1039481	259856	104163	25874	16.81	16.81	17.03	16.76
4	163734	40976	16260	4066	2.65	2.65	2.66	2.63
2	4925341	1231640	486095	123124	79.67	79.67	79.46	79.74

FIGURE 4 – Déséquilibre des classes dans le dataset

— Utilisation de l'option `handleInvalid="skip"` pour gérer l'apparition de nouvelles classes.
— Implémentation d'un **undersampling** des classes majoritaires (2 et 3).
— Ajout d'une pondération avec l'option `weightCol`.

Impact : les scores sont devenus plus justes (prédiction plus équilibrée entre les classes), mais les performances globales ont diminué (F1 ~0.5).

6. Sélection et optimisation du modèle
— Le modèle `LogisticRegression` a été retenu pour son équilibre entre précision et interprétabilité.
— Une recherche d'hyperparamètres a été réalisée via `GridSearch`.
— Le modèle optimisé a été réentraîné sur l'ensemble des données prétraitées.

3.3 Meilleurs modèles

Après avoir résolu la majorité des problèmes grâce aux tests effectués avec la régression logistique, notre objectif est désormais d'obtenir le meilleur modèle possible.

3.3.1 Équilibrage des données

Pour traiter le déséquilibre des classes, une stratégie d'échantillonnage a été appliquée :
— Échantillonnage aléatoire de 30000 instances par classe (cible)
— Conservation de toutes les instances pour les classes sous-représentées

En raison de limitations de mémoire RAM, il n'a pas été possible de travailler sur l'intégralité des données sans risquer un crash du système.

Sévérité	Nombre d'accidents (équilibré)
1	30,155
2	30,135
3	30,139
4	29,994

TABLE 1 – Distribution après équilibrage

3.4 Méthodologie

3.4.1 Algorithmes testés

Sept algorithmes de classification ont été évalués :

— Random Forest
— K-Nearest Neighbors (KNN)
— Régression Logistique
— Gradient Boosting
— AdaBoost
— Extra Trees
— Naive Bayes

3.4.2 Métriques d'évaluation

Les modèles ont été évalués selon quatre métriques :

— **Accuracy** : Proportion de prédictions correctes
— **F1-score macro** : Moyenne harmonique de la précision et du rappel
— **Précision macro** : Proportion de vrais positifs parmi les prédictions positives
— **Rappel macro** : Proportion de vrais positifs parmi les cas réellement positifs

3.4.3 Optimisation des hyperparamètres

Une recherche par grille (Grid Search) avec validation croisée 3-fold a été appliquée aux quatre meilleurs modèles pour optimiser leurs hyperparamètres.

3.5 Résultats

3.5.1 Comparaison initiale des modèles

Modèle	**Accuracy**	**F1-score**	**Précision**	**Rappel**
Random Forest	0.740	0.725	0.736	0.732
Gradient Boosting	0.736	0.720	0.725	0.726
Extra Trees	0.662	0.645	0.647	0.652
AdaBoost	0.626	0.611	0.614	0.621
KNN	0.484	0.467	0.481	0.470
Naive Bayes	0.442	0.350	0.484	0.420
Logistic Regression	0.388	0.313	0.341	0.361

TABLE 2 – Performance des modèles (configuration par défaut sans hyperparametre)

Voici la matrice de confusion pour certains modèles. On observe des performances globalement satisfaisantes, notamment pour le Gradient Boosting et le Random Forest. En revanche, d'autres modèles, comme le Naive Bayes, ont tendance à surajuster les données (overfitting), ce qui se traduit par des résultats moins fiables. :

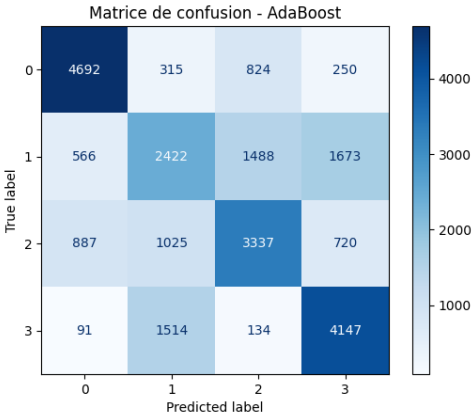


FIGURE 5 – (a) AdaBoost

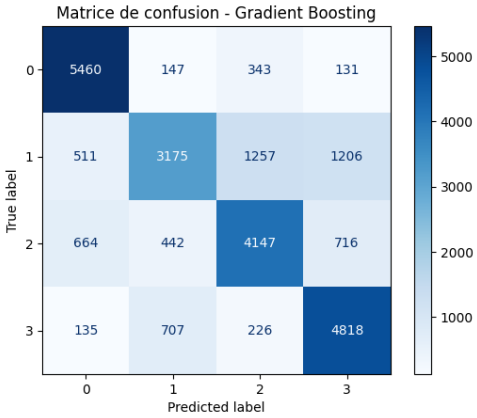


FIGURE 6 – (b) Gradient Boosting

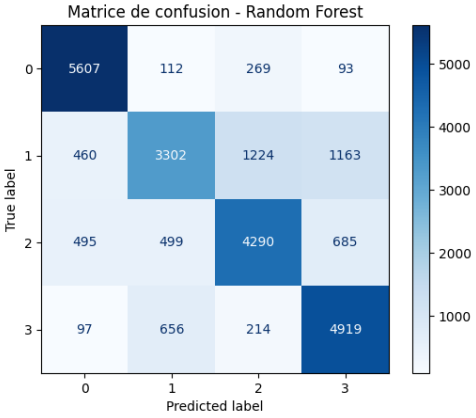


FIGURE 7 – (c) Random Forest

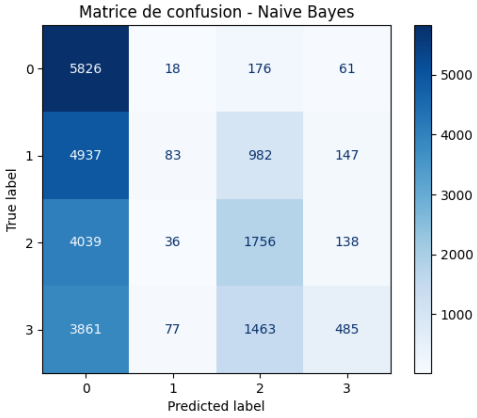


FIGURE 8 – (d) Naive Bayes

FIGURE 9 – Matrice de confusion pour les différents modèles de prédiction de la sévérité

3.5.2 Optimisation des hyperparamètres

Après optimisation par recherche par grille, les meilleurs paramètres obtenus sont :

Modèle	Score CV	Meilleurs paramètres
Gradient Boosting	0.700	learning_rate=0.1, max_depth=5, n_estimators=100
Random Forest	0.692	max_depth=20, n_estimators=200
Extra Trees	0.626	max_depth=20, n_estimators=200
AdaBoost	0.615	learning_rate=1.0, n_estimators=100

TABLE 3 – Résultats de l'optimisation des hyperparamètres

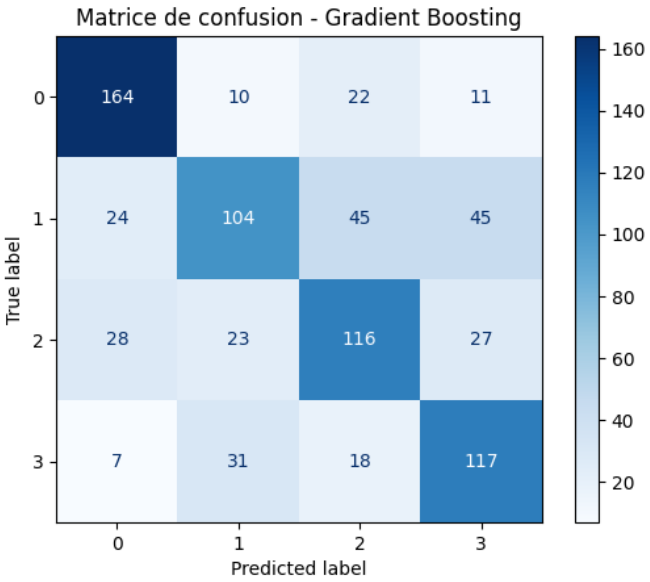
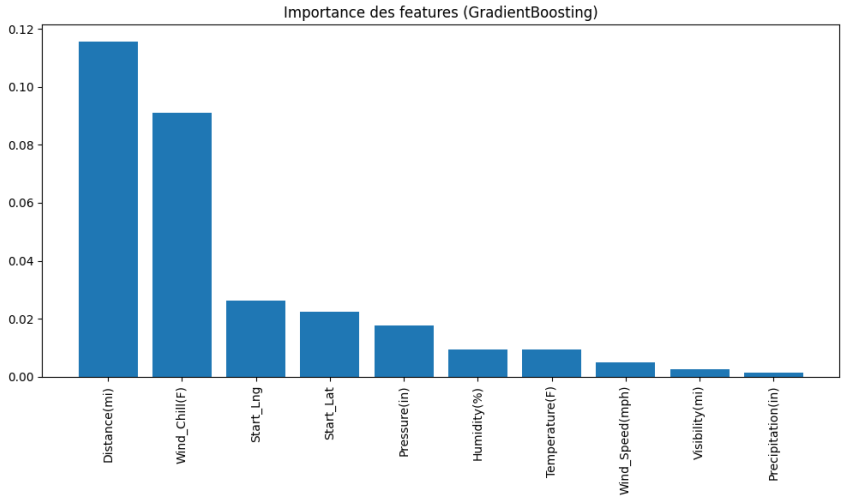
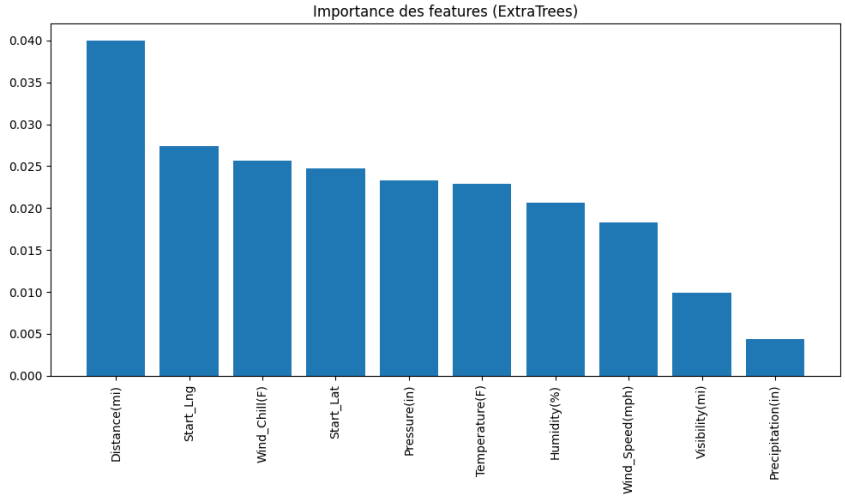
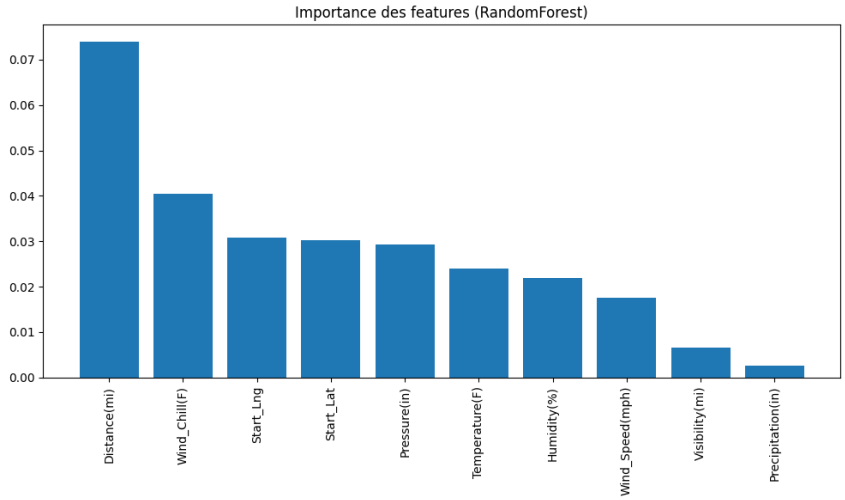


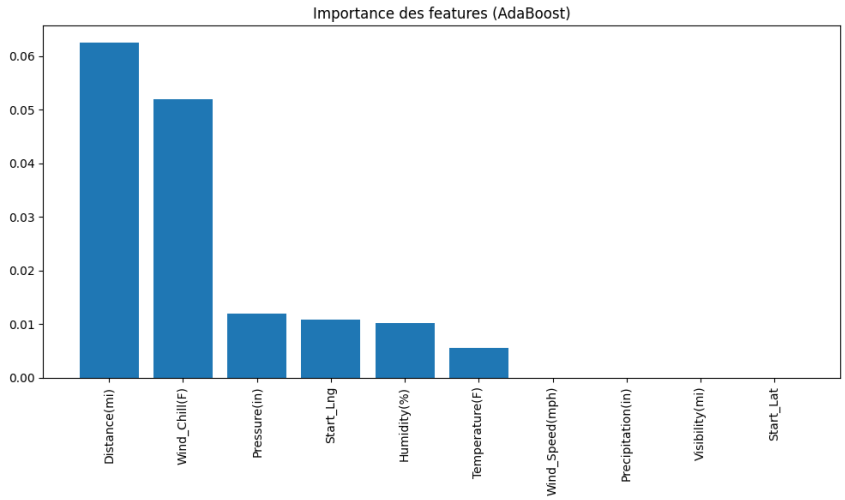
FIGURE 10 – Matrice de confusion du meilleurs modèle hyperparamètré

Le top 10 des variables les plus importantes a été identifié pour chacun des modèles. Il est notable que certaines variables apparaissent de manière récurrente, bien que leur importance varie d'un modèle à l'autre :









3.5.3 Performance du meilleur modèle

Le **Gradient Boosting** optimisé obtient les meilleures performances avec un score de validation croisée de 70.0%. Le rapport de classification détaillé montre :

Classe	Précision	Rappel	F1-score	Support
1	0.87	0.92	0.90	242
2	0.69	0.48	0.57	199
3	0.67	0.77	0.71	187
4	0.71	0.78	0.74	200
Macro avg	0.73	0.74	0.73	828
Weighted avg	0.74	0.75	0.74	828

TABLE 4 – Rapport de classification du Gradient Boosting optimisé

3.5.4 Analyse des résultats

Les résultats montrent que :

— Les modèles d'ensemble (Random Forest, Gradient Boosting) surpassent significativement les autres approches

— La classe 1 (sévérité minimale) est la mieux prédite avec 87% de précision et 92% de rappel

— La classe 2 présente le plus de difficultés avec seulement 48% de rappel

— L'accuracy globale de 75% représente une performance satisfaisante pour ce type de problème

3.5.5 Conclusion des modèles

Ces modèles démontre la faisabilité de la prédiction automatique de la sévérité des accidents de la route avec une précision de 75%. Le modèle Gradient Boosting optimisé constitue la meilleure approche testée, particulièrement performante pour identifier les accidents de faible sévérité.

Améliorations possibles :

— Techniques d'équilibrage plus sophistiquées (SMOTE, ADASYN)

— Analyse des caractéristiques les plus importantes

— Validation sur des données externes

— Exploration d'autres algorithmes (XGBoost, réseaux de neurones)

3.5.6 Limites et améliorations

Limites identifiées :

— Réduction drastique du dataset pour l'équilibrage des classes

— Performance variable selon les classes de sévérité

— Possible perte d'information lors de l'échantillonnage

4 Segmentation (clustering) des accidents par profil

Pour cette partie, nous avons voulu répondre à différentes questions en faisant apparaître plusieurs clusters sur notre jeu de données.

Puis nous avons voulu faire apparaître de nouveaux clusters en ajoutant des features/attributs après chaque mesure, nous permettant de rajouter du contexte tout en rajoutant des dimensions.

4.1 Jeu de données

4.1.1 Questions

Pour chaque question que nous nous sommes posées, nous avons effectué différentes limitations sur notre jeu de données , nous avons aussi fait des partitions en fonction du degré de gravité (`Severity`) de l'accident.

— Quel est la position du soleil ?

Attributs : 'Sunrise_Sunset_indexed','Severity'

clusters : 3 centroïdes

— Comment est la meteo ?

Attributs : 'Weather_Condition_indexed', 'Severity'

clusters : 4 centroïdes

— Quels sont les type de route les plus dangereuses ?

Attributs : 'Severity','Bump','Crossing','Give_Way','Junction','Railway','Roundabout','Station','Stop','

clusters : 4 centroïdes

PCA : Reduction en 9 dimension

— Quelles sont les variable atmospherique ?

Attributs : 'Severity','Temperature(F)','Wind_Chill(F)','Humidity(%)','Pressure(in)','Visibility(mi)','W

clusters : 5 centroïdes

PCA : Reduction en 6 dimension

4.1.2 Ajout d'attributs

Pour nos ajouts d'attributs nous avons 3 circonstances :

1. Situation initiale

   Attributs : 'Hour' et 'Severity'

   clusters : 3 centroïdes

2. Situation intermediaire

   Attributs Ajoutés : 'Temperature(F)' et 'Sunrise_Sunset_indexed'

   clusters : 3 centroïdes

3. Situation Finale

   Attributs Ajoutés : 'City_indexed' et 'Visibility(mi)'

   clusters : 5 centroïdes

   PCA : Reduction en 6 dimension

4.2 Méthode

4.2.1 Questions

Nous suivons ce processus :

1. Enumeration de la question,des attributs concernés, du nombre de cluster et de `PCA`.
2. Limitation sur le jeu de données original en fonction des attributs concernés.
3. Partition du jeu de données limités en fonction du degrée de gravité , chaque partition est une condition.

   1^{ere} partition : Ce jeu de données ne représente que les accidents mineur (le degrée de gravité le plus bas).

   2^{eme} partition : Ce jeu de données ne s'intèresse qu'au accident majeur (le degrée de gravité le plus haut).

   3^{eme} partition : tout les degrées sont comptabilisée.

puis pour chaque partition :

1. Nous assemblons les colonnes d'entrée en un vecteur de caractéristiques à l'aide de `VectorAssembler`, puis nous appliquons une mise à l'échelle (`Scaling`) standardisée des données.
2. Si necessaire, nous réduisons la dimension des données à l'aide de l'analyse en composantes principales (`PCA`).
3. Nous appliquons un modèle de classification non supervisée `K-Means` afin de regrouper les observations en clusters.
4. Nous transformons les centroïdes des clusters du jeu de données réduit à l'espace d'origine, en inversant les transformations de `PCA` (le cas échéant) et de mise à l'échelle.
5. Nous plottons les centroïdes du jeu de données réduit afin d'avoir un contexte visuel.
6. Nous convertissons les résultats en un tableau `pandas` dans le but de transformer les données qui ont été indexes durant le `pre-processing` dans leur forme d'origine, en les réassignant.

Ainsi nous récupérons les attributs concernés puis nous faisons des clusters sur les données en `PCA` et transformons ces clusters en données originales et en tirons nos conclusions.

4.2.2 Ajout d'attributs

Nous suivons le même processus cependant au lieu de repartir de zéro pour les attributs concernés, nous les rajoutons avant de partitionner.

4.3 Analyses

Pour chaque question nous allons vous présenter une représentation visuelle et une table avec les valeurs de nos centroïdes en fonction des attributs et des partitions et faire une conclusion intermédiaire.

4.3.1 Questions

Pour la question 'Quel est la position du soleil ?' nous avons ces différents clusters :

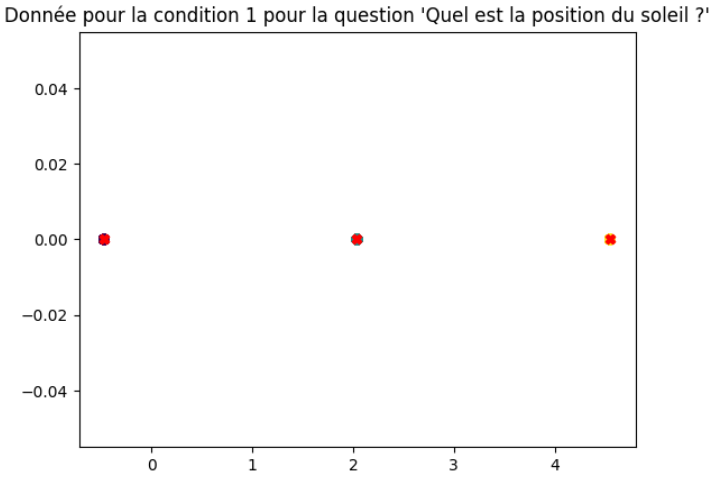


FIGURE 11 – Centroïdes des données sur la condition 1 : les accidents les plus mineurs

Sunrise_Sunset	Severity
Day	1.0
Night	1.0
Unknown	1.0

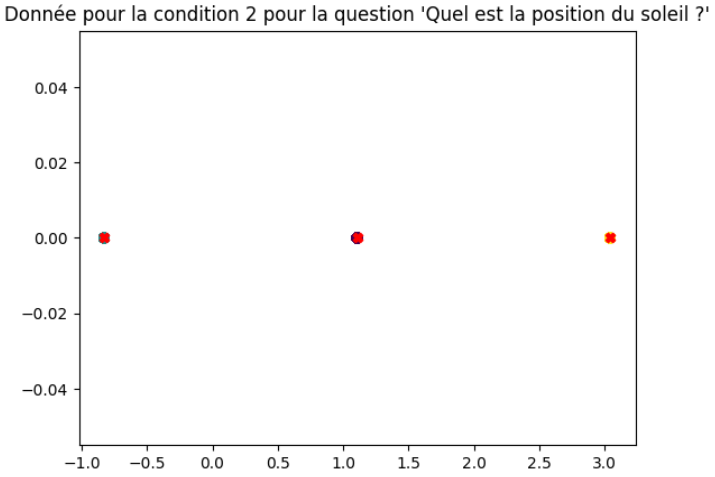


FIGURE 12 – Centroïdes des données sur la condition 2 : les accidents les plus majeurs

Sunrise_Sunset	Severity
Night	4.0
Day	4.0
Unknown	4.0

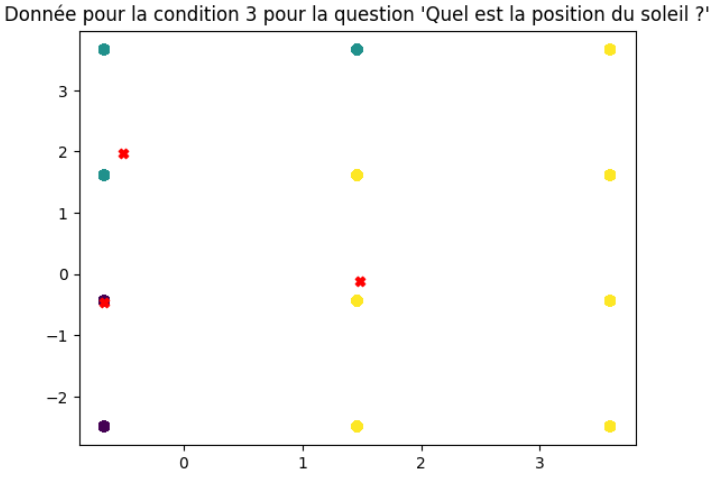


FIGURE 13 – Centroïdes des données sur la condition 3 : Tout les accidents

Sunrise_Sunset	**Severity**
Day	1.987176
Day	3.178263
Night	2.155040

Comme vous pouvez le remarquer la représentation des clusters pour les accidents mineurs et majeurs ne nous donnes que peu d'information. Cependant lorsque nous prenons en compte chaque accident nous pouvons remarquer que la majeur partie de nos accident se déroule lorsque le soleil est apparent. possiblement expliqué par la période la plus fréquentée de la journée.

Pour la question 'Comment est la meteo ?' nous avons ces different cluster :

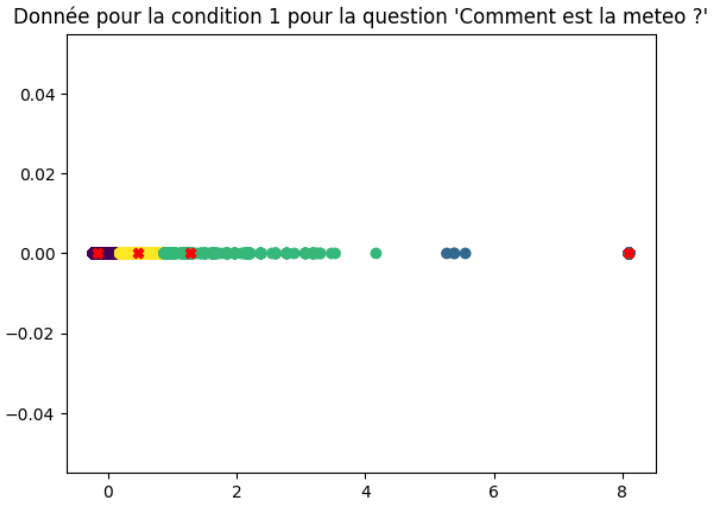


FIGURE 14 – Centroïdes des données sur la condition 1 : les accidents les plus mineurs

Weather_Condition	Severity
Mostly Cloudy	1.0
Unknown	1.0
Light Rain / Windy	1.0
Fair / Windy	1.0

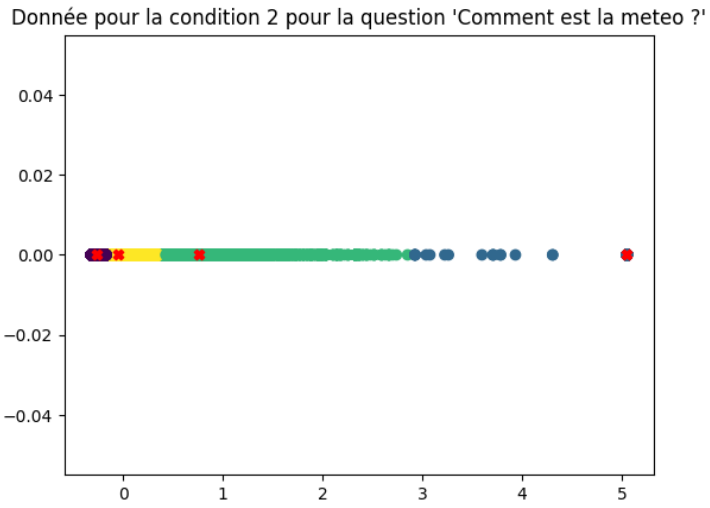


FIGURE 15 – Centroïdes des données sur la condition 2 : les accidents les plus majeurs

Weather_Condition	Severity
Mostly Cloudy	4.0
Unknown	4.0
Light Thunderstorms and Rain	4.0
Scattered Clouds	4.0

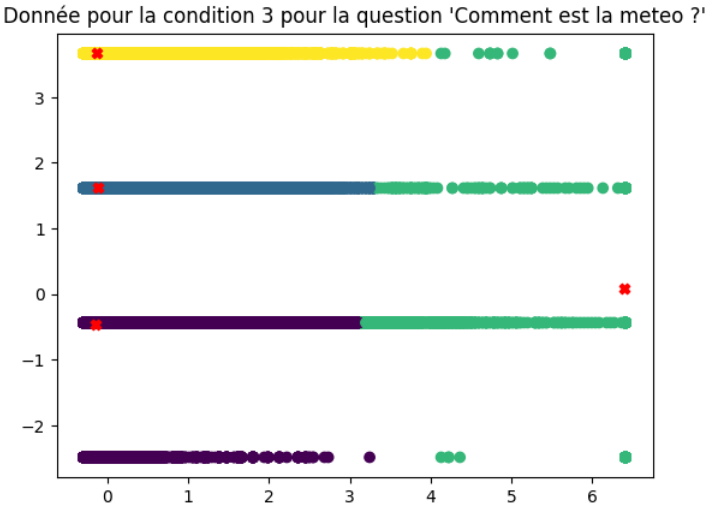


FIGURE 16 – Centroïdes des données sur la condition 3 : Tout les accidents

Nous pouvons en conclure que la plupart des accident mineurs apparaissent quand du vent se fait ressentir et lorsque des nuages se forment alors si un accident se produit, le degrée de gravité peut augmenter. La derniere condition nous permet de comprendre que beaucoup d'accident se

Weather_Condition	Severity
Clear	1.989092
Partly Cloudy	3.000000
Unknown	2.254228
Clear	4.000000

produisent lors d'un ciel bleue et que cet attribut ne nous permet pas d'evaluer le degrée mais il met en place les condition pour qu'un accident se produit.

Pour la question 'Quels sont les type de route les plus dangereuses ?' nous avons ces differents cluster :

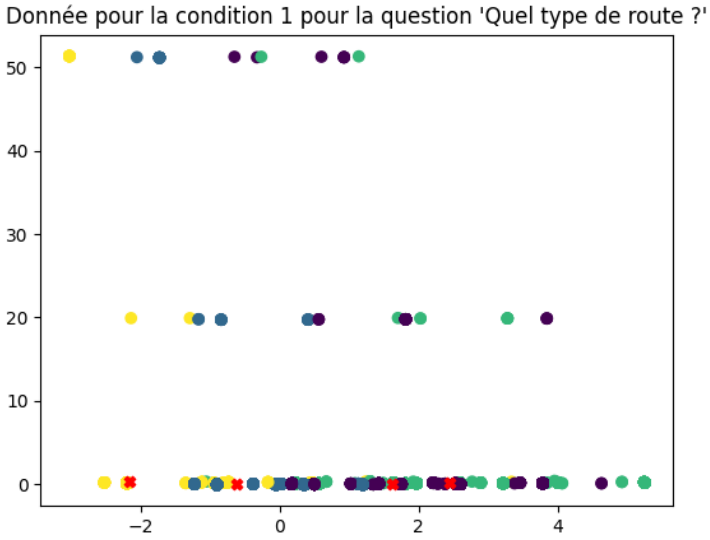


FIGURE 17 – Centroïdes des données sur la condition 1 : les accidents les plus mineurs

Severity	Bump	Crossing	Give_Way	Junction	Railway
1.0	4.13e-04	9.996e-01	1.63e-02	-7.47e-15	4.56e-02
1.0	3.36e-04	9.61e-14	5.90e-03	2.23e-14	8.97e-04
1.0	7.40e-04	6.34e-01	1.48e-03	4.07e-03	8.92e-02
1.0	3.21e-03	1.48e-02	1.93e-03	1.00e+00	3.21e-03

Roundabout	Station	Stop	Traffic_Calming	Traffic_Signal	Turning_Loop
7.12e-17	1.77e-14	4.80e-02	2.01e-03	8.22e-01	0.0
-2.00e-17	-2.41e-15	3.73e-02	6.95e-04	2.37e-01	0.0
-4.47e-16	1.00e+00	5.18e-02	3.33e-03	6.70e-01	0.0
2.87e-16	2.25e-03	7.70e-03	3.85e-03	1.09e-02	0.0

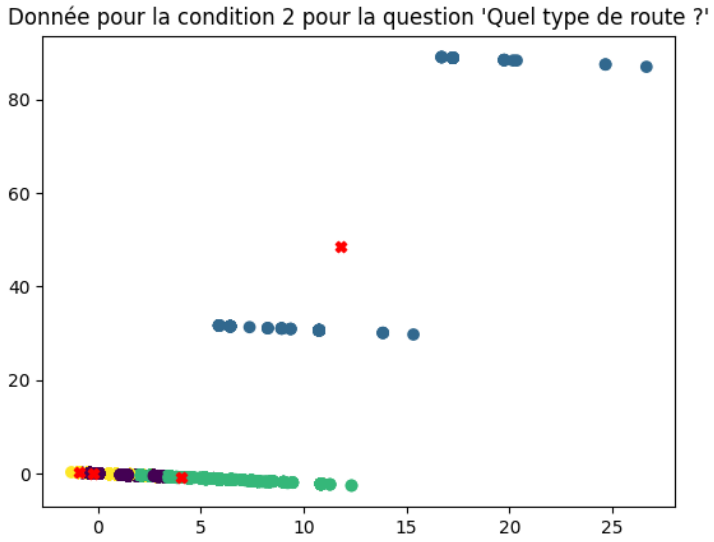


FIGURE 18 – Centroïdes des données sur la condition 2 : les accidents les plus majeurs

Severity	Bump	Crossing	Give_Way	Junction	Railway
4.0	1.39e-04	6.50e-04	4.69e-03	-8.10e-08	-9.40e-05
4.0	4.21e-01	1.03e+00	2.54e-02	1.41e-01	-7.00e-02
4.0	-2.42e-03	9.43e-01	2.21e-02	1.08e-02	1.10e-01
4.0	-7.60e-05	4.88e-03	2.99e-03	1.00e+00	2.35e-03

Roundabout	Station	Stop	Traffic_Calming	Traffic_Signal	Turning_Loop
2.94e-05	5.17e-03	2.75e-02	-2.54e-04	6.44e-02	0.0
4.06e-05	5.93e-02	-8.37e-03	7.84e-01	-9.55e-02	0.0
-8.29e-07	1.33e-01	6.63e-02	4.42e-03	6.50e-01	0.0
4.90e-05	1.43e-03	3.56e-03	1.40e-04	5.53e-03	0.0

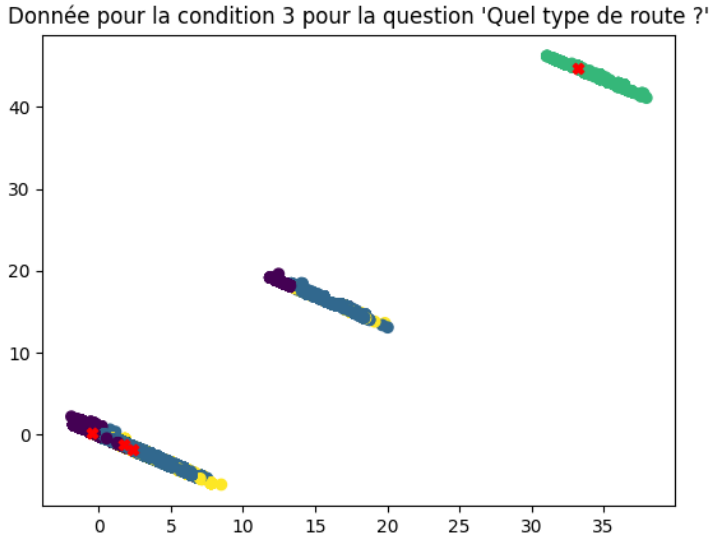


FIGURE 19 – Centroïdes des données sur la condition 3 : Tout les accidents

Severity	Bump	Crossing	Give_Way	Junction	Railway
1.0	4.21e-04	9.77e-01	1.52e-02	8.00e-06	-1.71e-06
1.0	3.43e-03	1.55e-02	1.93e-03	9.9998e-01	4.92e-06
1.0	3.51e-04	1.00e-05	5.85e-03	-2.00e-06	4.17e-07
1.0	-1.32e-04	9.17e-01	3.76e-03	1.41e-02	1.00e+00

Roundabout	Station	Stop	Traffic_Calming	Traffic_Signal	Turning_Loop
-2.31e-17	1.05e-01	4.72e-02	2.72e-03	8.35e-01	0.0
2.84e-16	4.19e-03	8.03e-03	3.51e-03	1.12e-02	0.0
-2.66e-17	1.23e-02	3.80e-02	5.02e-04	2.34e-01	0.0
6.90e-16	2.26e-01	4.61e-02	3.03e-03	6.03e-01	0.0

Grace a cette question nous pouvons comprendre dans quelle circonstance les accident se produisent. En remarquant les valeurs très faible de chaque attribut, nous pouvons conclure que la plupart des accident se produisent sur des routes droites ou qui ne possèdent pas de signalisation définie par notre jeu de donnée.

Pour la question 'Quelles sont les variable atmospherique ?' nous avons ces differents cluster :

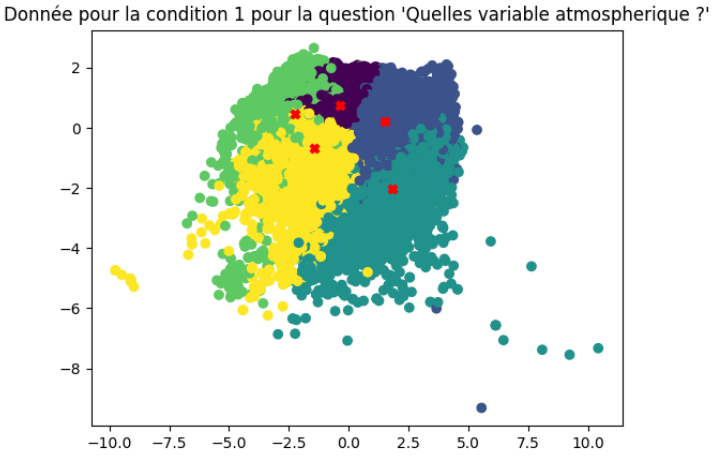


FIGURE 20 – Centroïdes des données sur la condition 1 : les accidents les plus mineurs

Severity	Temperature (F)	Wind_Chill (F)	Humidity (%)	Pressure (in)	Visibility (mi)	Wind_Speed (mph)
1.0	73.684810	73.418444	79.860183	29.588027	9.783071	4.323339
1.0	84.292530	83.989659	40.953438	29.327295	9.974908	9.926147
1.0	77.446520	77.258828	23.613301	26.296171	10.049023	8.702263
1.0	62.218820	61.649029	91.825701	29.392472	3.339218	6.213618
1.0	56.177776	55.543673	67.992956	29.577558	9.875727	7.525184

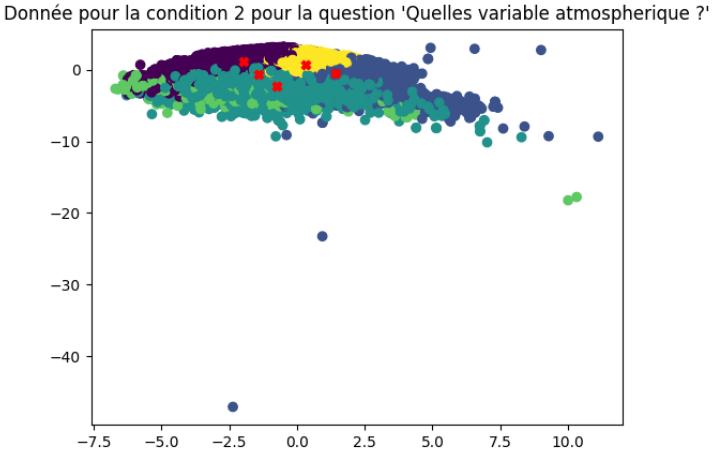


FIGURE 21 – Centroïdes des données sur la condition 2 : les accidents les plus majeurs

Severity	Temperature (F)	Wind_Chill (F)	Humidity (%)	Pressure (in)	Visibility (mi)	Wind_Speed (mph)
4.0	44.641273	43.183162	91.915521	29.508907	2.673912	7.799399
4.0	74.372468	70.465468	44.903793	29.605668	10.078388	9.358777
4.0	47.838948	44.911248	52.893900	24.351585	9.280888	8.690655
4.0	34.574331	30.214795	67.234078	29.659869	9.837304	8.416086
4.0	64.109950	63.568992	79.718281	29.716552	9.674743	5.551549

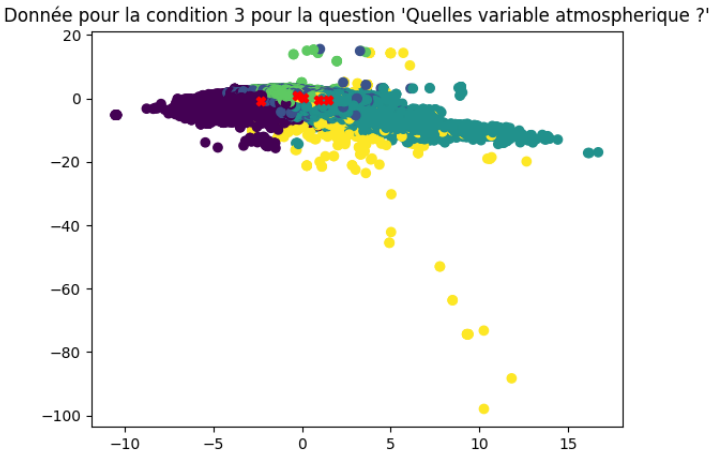


FIGURE 22 – Centroïdes des données sur la condition 3 : Tout les accidents

Severity	Temperature (F)	Wind_Chill (F)	Humidity (%)	Pressure (in)	Visibility (mi)	Wind_Speed (mph)
2.138	30.823	27.210	73.856	29.018	7.835	8.699
3.157	64.782	61.717	66.958	29.779	9.207	7.524
2.024	76.585	73.923	38.153	29.223	10.071	6.425
1.987	61.560	60.597	79.354	29.790	8.779	4.880
2.021	71.866	68.117	57.458	29.754	9.805	14.126

Grace a cette question nous pouvons remarquer que la visibilité , la temperature et la vitesse du vent joue une énorme importance pour detecter le degrée de gravité de la situation.

4.3.2 Ajouts d'attributs

Pour ces ajouts d'attribut nous possedons comme representation des centroides :

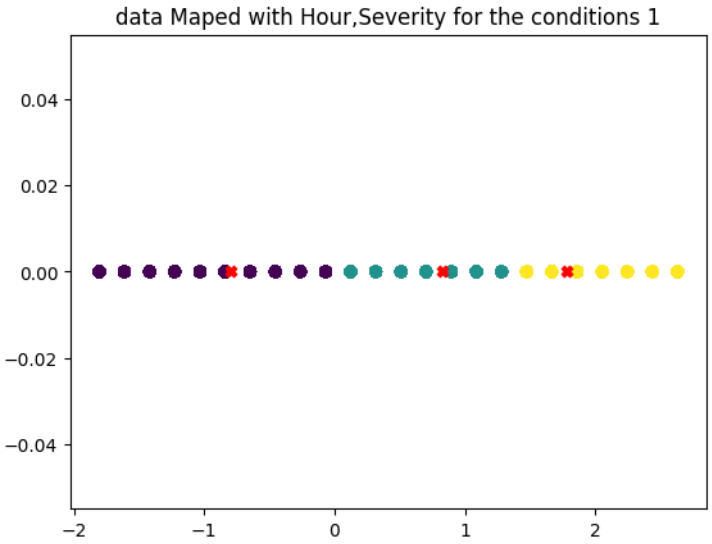


FIGURE 23 – Centroïdes des données sur les attributs 'Hour' et 'Severity' condition 1

Hour	**Severity**
5.214	1.0
13.630	1.0
18.582	1.0

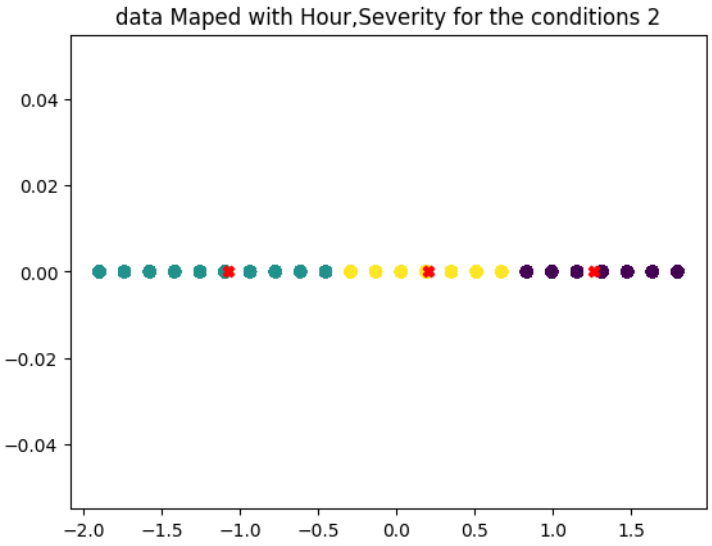


FIGURE 24 – Centroïdes des données sur les attributs 'Hour' et 'Severity' condition 2

Hour	**Severity**
19.685	4.0
5.125	4.0
13.095	4.0

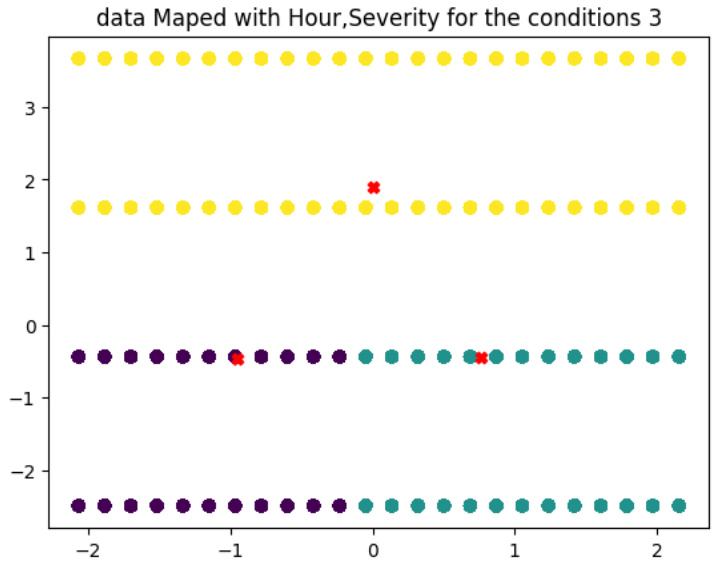


FIGURE 25 – Centroïdes des données sur les attributs 'Hour' et 'Severity' condition 3

Hour	Severity
6.085	1.986
15.438	1.992
11.277	3.136

Nous pouvons remarquer que l'heure de l'accident ne permet pas de detecter le degrée de gravité cependant nous remarquons que la plupart des accidents se produisent au alentours de 5h du matin et 13h de l'apres-midi

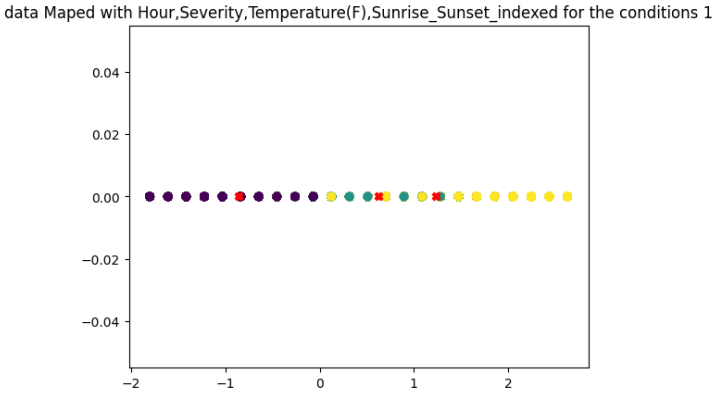


FIGURE 26 – Centroïdes des données en rajoutant les attributs 'Temperature(F)' et 'Sunrise_Sunset_indexed' condition 1

Hour	Severity	Temperature (F)	Sunrise_Sunset
4.892	1.0	67.275	Day
12.572	1.0	84.819	Day
15.777	1.0	60.464	Day

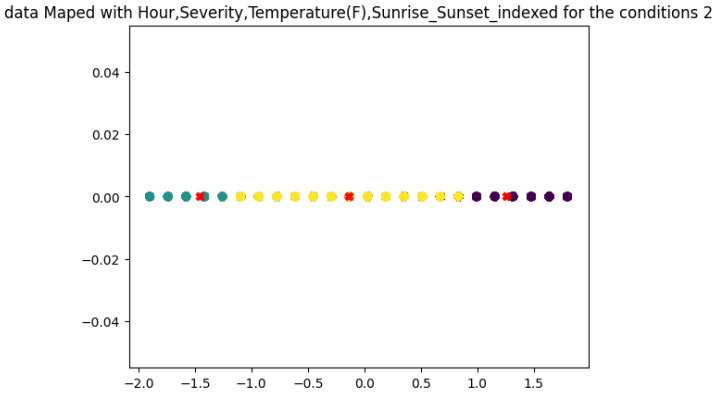


FIGURE 27 – Centroïdes des données en rajoutant les attributs 'Temperature(F)' et 'Sunrise_Sunset_indexed' condition 2

Hour	Severity	Temperature (F)	Sunrise_Sunset
19.630	4.0	53.840	Night
2.758	4.0	49.506	Night
10.973	4.0	62.769	Day

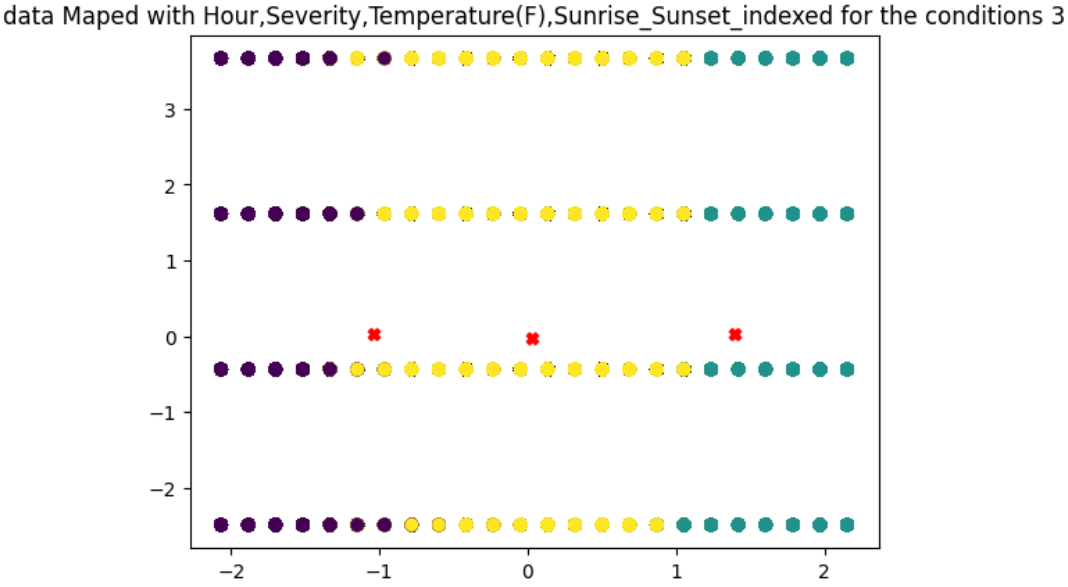


FIGURE 28 – Centroïdes des données en rajoutant les attributs 'Temperature(F)' et 'Sunrise_Sunset_indexed' condition 3

Hour	Severity	Temperature (F)	Sunrise_Sunset
5.627	2.224	44.609	Day
18.888	2.223	55.058	Night
11.410	2.203	71.989	Day

Nous pouvons remarquer que la plupart des accident grave se produisent le soir avec une temperature se trouvant dans les 50 Fahrenheit tandis que les accident mineurs se produise plus le jour lorsque la temperature avoisinent les 70 Fahrenheit.

Comme expliqué précédemment aux alentours de 5h la plupart des accidents se produisent mais la température nous permet d'être plus précis.

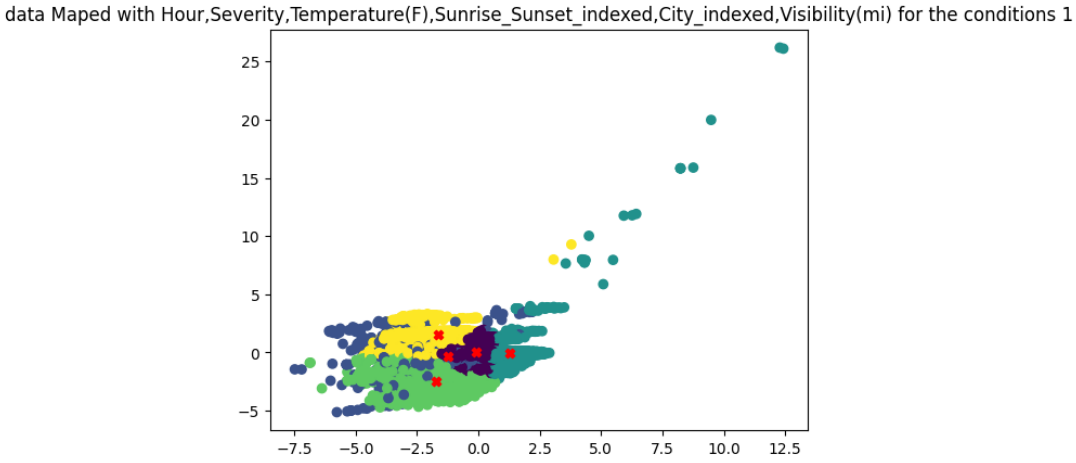


FIGURE 29 – Centroïdes des données finale, ajout des attributs 'City_indexed' et 'Visibility(mi)' condition 1

Hour	**Severity**	**Temperature (F)**	**Sunrise_Sunset**	**City**	**Visibility (mi)**
9.073	1.0	71.507	Day	South El Monte	9.372
8.274	1.0	64.816	Day	Montesano	9.914
12.754	1.0	82.570	Day	San Mateo	10.269
6.557	1.0	55.059	Day	Saint Augustine	4.171
3.390	1.0	60.024	Night	Encino	10.238

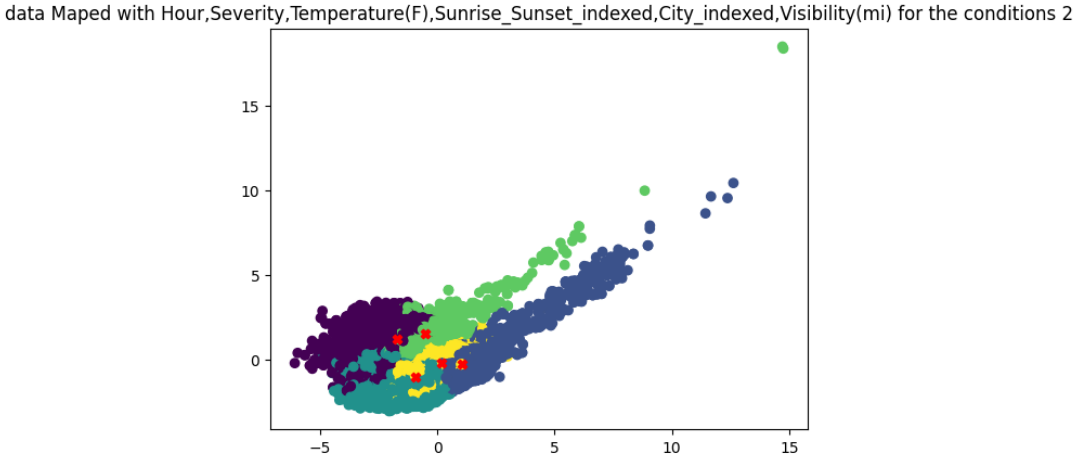


FIGURE 30 – Centroïdes des données finale, ajout des attributs 'City_indexed' et 'Visibility(mi)' condition 2

Hour	Severity	Temperature (F)	Sunrise_Sunset	City	Visibility (mi)
17.447	4.0	41.105	Night	Bechtelsville	8.702
10.388	4.0	71.137	Day	Sebastopol	10.057
7.293	4.0	42.150	Night	Wytheville	6.562
19.121	4.0	54.681	Night	Parkville	10.073
10.492	4.0	62.586	Day	Peterborough	9.499

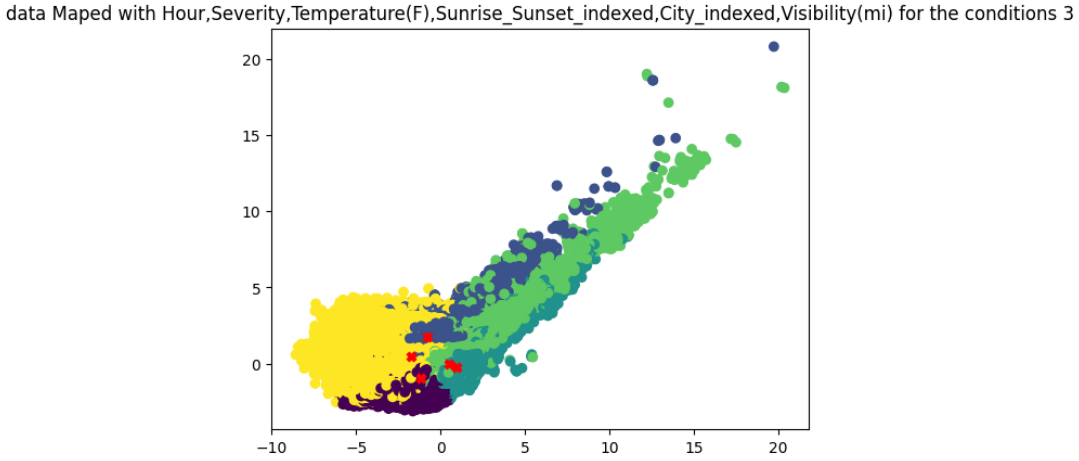


FIGURE 31 – Centroïdes des données finale, ajout des attributs 'City_indexed' et 'Visibility(mi)' condition 3

Hour	Severity	Temperature (F)	Sunrise_Sunset	City	Visibility (mi)
7.114	2.053	44.684	Day	City Of Industry	6.736
18.801	2.092	53.879	Night	Abingdon	9.868
10.683	2.044	72.263	Day	Fremont	9.805
10.591	2.763	70.199	Day	South Saint Paul	10.022
11.708	3.422	46.033	Night	Fort Monroe	8.549

Le rajout de la visibilité et de la ville nous permet de voir une accumulation d'accident aux environ de 11h avec de forte temperature malgré une bonne visibilité.

4.4 conclusion

Nous avons découvert successivement plusieurs questions par K-Means sur différentes combinaisons d'attributs, en partitionnant le jeu de données selon le degré de gravité. Dans le but d'analyser nos données nous avons effectué plusieurs formats :

— un tableau pandas avec les valeurs de nos centroïdes basés sur les attributs originaux

— une visualisation de nos cluster en utilisant la réduction de dimension

Nous avons aussi observé l'ajout progressif de nouvelles dimensions (attribut) et montré l'utilité de la réduction de dimension (PCA) pour la stabilité et la lisibilité des clusters. Elle permet aussi de réussir notre détection de cluster, sans cette réduction notre machine ne peut supporter les calculs et ne nous donnes pas de résultats avec l'utilisation de plus de 4 attributs.

Nous avons remarquer aussi la prédominance d'accidents de degrée 2 rendant le clustering biaisée sur des circonstance d'accident normaux, la partition en accident mineur et majeur permettait de voir les donnée significative en fonction du degrée de gravité .

Concernant nous données nous avons découvert :

— Les accidents, mineurs se produisent particulièrement en plein jour et les accident majeurs la nuit, mais sur l'ensemble des données, la majorité se produit en plein jour.
— Les accidents mineurs surviennent surtout par vent modéré. Les accidents graves s'associent souvent avec les orages légers et une visibilité réduite. La visibilité et la force du vent constituent des donnée plus significative que les précipitations.
— Les coefficients faibles tel que 'Bump', 'Crossing' ou 'Junction' indiquent que la plupart des accidents surviennent sur des routes simples tel que des routes droites.
— La visibilité et la température désignent les plus grandes variations entre les degrée de gravité mineurs et majeurs . Les vents forts et humidité élevée se présentent lors des cas de gravité grave.
— Avec l'heure seule, les clusters restent peu interprétables, hormis une légère concentration à l'aube (5h) et en début d'après-midi (13h).

5 Détection automatique d'accidents anormaux à partir de données météorologiques, temporelles et spatiales

5.1 Objectif du projet

Ce projet a pour but d'identifier automatiquement des accidents de la route considérés comme anormaux en se basant sur des données météorologiques, temporelles et géographiques. Il s'inscrit dans une logique de prévention et d'analyse des risques en identifiant des événements atypiques non perceptibles via une simple analyse statistique descriptive.

Les anomalies peuvent correspondre, par exemple, à des accidents survenus dans des conditions météorologiques clémentes, à des horaires peu fréquents ou dans des zones historiquement peu sujettes aux accidents.

5.2 Traitement et échantillonnage des données

Les données, initialement volumineuses et stockées au format `Parquet`, ont été chargées via Apache Spark pour leur traitement distribué. Après une sélection des colonnes pertinentes (gravité, localisation, météo, heure, etc.), elles ont été converties en `DataFrame` Pandas afin d'être utilisées dans des algorithmes de machine learning tels que `Isolation Forest` et `HDBSCAN`.

Remarque importante : Nous avons volontairement choisi de ne travailler que sur **10%** de l'échantillon total. Cette décision s'explique par des limitations matérielles : le traitement complet du jeu de données entraînait des durées de calcul excessives, et dans le cas de certains modèles (comme `DBSCAN`), des dépassements mémoire voire des blocages.

5.3 Choix des algorithmes et justification

Plusieurs algorithmes de détection d'anomalies non supervisée ont été évalués :

— `Isolation Forest`, pour sa robustesse et son efficacité sur de grands volumes de données.
— `Local Outlier Factor (LOF)`, pour ses capacités à détecter des anomalies locales.

— `KMeans`, afin de disposer d'un clustering baseline.
— `HDBSCAN`, pour sa capacité à détecter des clusters de densité variable sans avoir à spécifier le nombre de clusters.

Nous avons initialement envisagé l'utilisation de `DBSCAN`. Cependant, en raison des performances médiocres observées (temps de calcul très élevés, mauvaise gestion du bruit avec de grands volumes de données), ce dernier a été abandonné au profit de `HDBSCAN`, mieux adapté à notre problématique.

5.4 Exploration visuelle et analyse contextuelle

Avant toute modélisation, une phase d'analyse exploratoire a permis de mieux comprendre la structure des données. Les visualisations suivantes ont été réalisées :

— Cartographie des accidents par gravité sur le territoire américain (via `Plotly` et `Folium`).
— Histogramme des niveaux de gravité (`Severity`).
— Distribution des accidents par jour de la semaine.
— Analyse de l'influence des variables météo (température, humidité, visibilité).

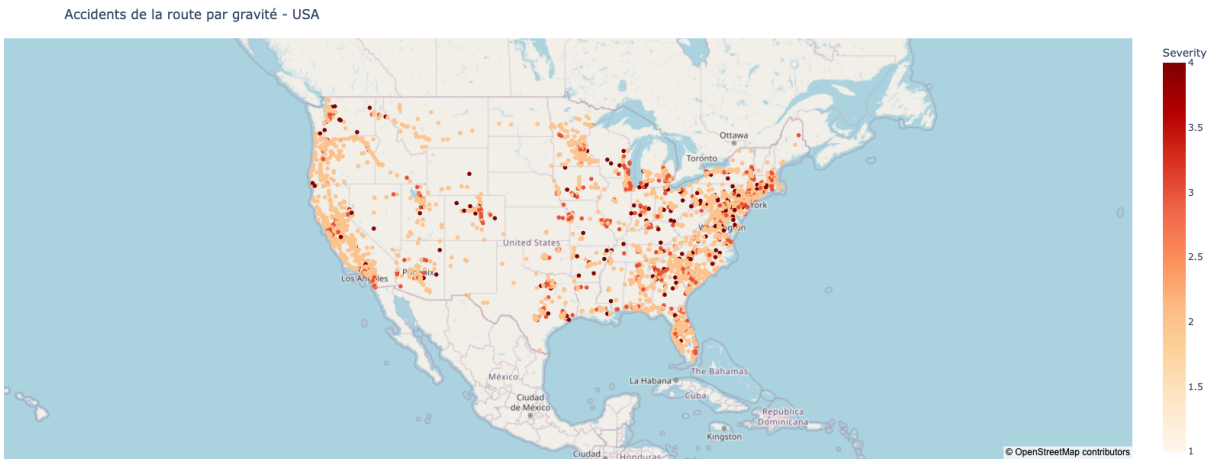


FIGURE 32 – Carte des accidents par gravité (USA)

Ces visualisations confirment l'intérêt d'une détection d'anomalies : la majorité des accidents se concentrent dans les grandes métropoles et aux horaires prévisibles. Un accident survenant dans un désert à 3h du matin, sous une météo clémente, constitue un bon candidat à l'anomalie.

5.5 Prétraitement et encodage

Le prétraitement des données a été une étape cruciale, compte tenu du volume du dataset et de la diversité des variables disponibles (temporelles, météorologiques, géographiques, et structurelles). L'objectif était de rendre les données exploitables par des algorithmes de détection d'anomalies, tout en préservant leur cohérence et leur richesse informative.

Sélection des variables pertinentes

Toutes les colonnes disponibles dans le dataset brut n'étaient pas pertinentes pour l'analyse. Un sous-ensemble de variables a été sélectionné selon les critères suivants :

- **Pertinence théorique** : les colonnes comme `Weather_Condition`, `Visibility`, `Start_Time`, ou `Start_Lat/Start_Lng` ont un lien potentiel avec le caractère inhabituel d'un accident.
- **Taux de valeurs manquantes** : les variables trop incomplètes ont été écartées pour éviter un biais ou une perte de données excessive.
- **Type de variable** : priorité donnée aux variables numériques ou convertibles (catégorielles avec peu de modalités).

Encodage des variables catégorielles

Certaines colonnes, bien que catégorielles (ex. : `Weather_Condition`, `Wind_Direction`, `State`, etc.), contenaient une information riche et exploitable. Trois techniques ont été considérées :

- `LabelEncoder` : utilisé pour les colonnes ordinales ou peu modales.
- `OneHotEncoder` : envisagé mais peu adapté ici à cause de la dimensionnalité (plusieurs milliers de villes, directions de vent variées, etc.).
- **Encodage vectoriel indexé** : méthode privilégiée ici, consistant à représenter les catégories par des vecteurs creux (`VectorIndexer`), compatible avec Spark et HDBSCAN.

Cette approche a permis de limiter la taille de l'espace vectoriel tout en conservant les distinctions utiles entre catégories.

Normalisation des variables continues

Afin d'éviter que des variables numériques à forte amplitude (ex. : `Distance`, `Temperature`) dominent les calculs de distance ou d'inertie dans les modèles comme `KMeans`, une standardisation (`StandardScaler`) a été appliquée.

Sous-échantillonnage raisonné

Le jeu de données initial comprenait plusieurs millions d'observations. Or, certains algorithmes comme `DBSCAN` ou `LOF` n'étaient pas exécutables à cette échelle sur notre machine. Un échantillon aléatoire de **10%** a donc été extrait avec `Spark.sample()`, en garantissant :

- Une conservation de la distribution des niveaux de gravité,
- Un équilibre représentatif sur les dimensions temporelles et spatiales,
- Un compromis entre performance et fiabilité.

Évaluation de la redondance

Pour éviter d'introduire des variables fortement corrélées entre elles (ce qui nuit à la stabilité des modèles), une matrice de corrélation a été générée. Celle-ci a servi à identifier les dépendances linéaires et à ajuster les variables finales en entrée.

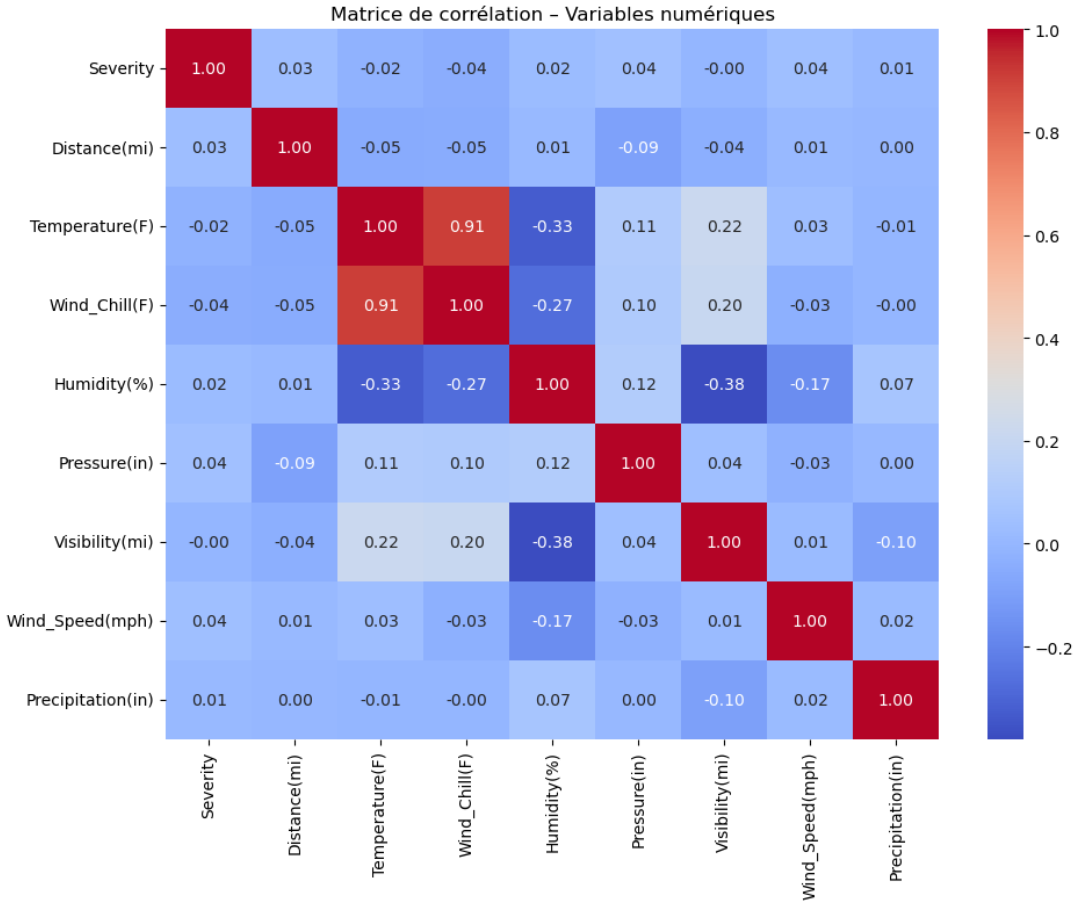


FIGURE 33 – Matrice de corrélation des variables numériques

5.6 Conclusion intermédiaire

Le pipeline mis en place repose sur :

— Un sous-échantillonnage de 10% du dataset original,

— Un nettoyage et encodage des variables pertinentes,

— Des visualisations pour interpréter les distributions et justifier l'approche,

— Le recours à `HDBSCAN` pour la robustesse et la flexibilité dans l'identification d'accidents atypiques.

L'étape suivante a consisté à entraîner et comparer différents modèles pour extraire les anomalies significatives du corpus. Ces résultats feront l'objet de la section suivante.

5.7 Détection d'anomalies : méthodes appliquées et résultats observés

Pour identifier les accidents anormaux, quatre modèles non supervisés ont été appliqués sur les données traitées. L'objectif était de repérer des événements rares ou atypiques. Voici un aperçu détaillé de chaque méthode, des anomalies qu'elle a révélées, et des observations associées.

1. Isolation Forest

`Isolation Forest` est basé sur le principe que les anomalies sont plus facilement isolables. Il construit des arbres de partition aléatoires pour détecter les observations inhabituelles.

- **Nombre d'anomalies détectées :** 7 730
- **Nombre d'observations normales :** 765 187
- **Anomalies typiques :**
  - Accidents dans des zones rurales ou à faible trafic,
  - Conditions météo modérées où un accident paraît inattendu,
  - Horaires inhabituels (nuit, week-ends, heures creuses).
- **Avantages :** traitement rapide, bonne détection globale.
- **Limites :** peu explicable, nécessite de fixer un seuil arbitraire.

2. Local Outlier Factor (LOF)

`LOF` identifie des anomalies à partir de la densité locale des points. Une observation est considérée comme anormale si elle se trouve dans une région peu dense par rapport à ses voisins.

- **Nombre d'anomalies détectées :** 7 730
- **Nombre d'observations normales :** 765 187
- **Anomalies typiques :**
  - Points isolés dans l'espace,
  - Cas avec des valeurs extrêmes ou absentes dans certaines variables,
  - Situations météorologiques très rares.
- **Avantages :** pertinence locale.
- **Limites :** sensible au bruit, plus lent que `Isolation Forest`.

3. KMeans (distance au centre)

Bien que `KMeans` soit initialement un algorithme de clustering, il a été utilisé ici comme base de comparaison pour la détection d'anomalies. L'idée repose sur le fait que les points très éloignés des centres des clusters (centroïdes) peuvent être considérés comme atypiques.

- **Méthodologie :**
  - Le nombre de clusters k a été déterminé grâce à la **méthode du coude** (Elbow method), qui consiste à tracer l'inertie intra-cluster en fonction de k,
  - Une valeur de k est retenue lorsque l'amélioration de l'inertie devient marginale (forme de coude sur la courbe),
  - Une fois les clusters formés, la distance de chaque observation à son centroïde a été calculée : les plus éloignées sont considérées comme des anomalies.
- **Anomalies typiques :**
  - Observations situées en périphérie des regroupements principaux,
  - Accidents dans des zones géographiques peu représentées.
- **Limites :**
  - L'algorithme suppose des clusters sphériques et de taille similaire,
  - Sensible aux outliers, et nécessite de spécifier k a priori.

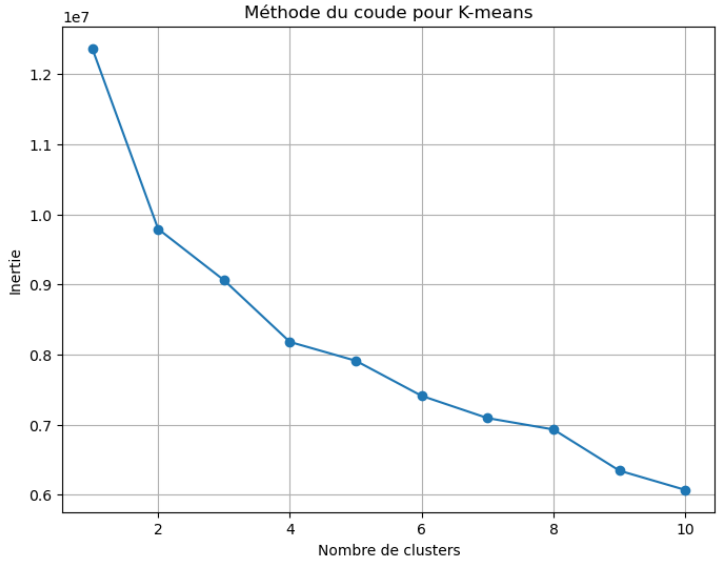


FIGURE 34 – Méthode du coude pour choisir le nombre optimal de clusters k

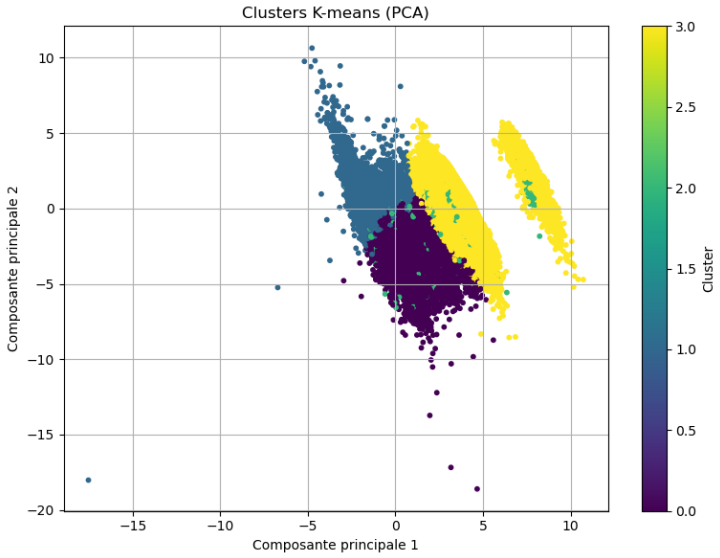


FIGURE 35 – Projection PCA des anomalies détectées par KMeans

4. HDBSCAN

`HDBSCAN` a permis une détection efficace du bruit (outliers) en identifiant automatiquement des clusters de densité variable.

- **Nombre d'anomalies détectées (points marqués comme bruit) :** 127 885
- **Nombre d'observations assignées à un cluster :** 645 032
- **Anomalies typiques :**
  - Accidents isolés dans le temps ou l'espace,
  - Contextes météo incohérents par rapport au lieu,
  - Faibles densités locales non intégrées dans des groupes.
- **Avantages :** pas besoin de fixer le nombre de clusters, filtrage automatique du bruit.

— **Limites :** plus lent à l'exécution, sensible au paramètre `min_samples`.

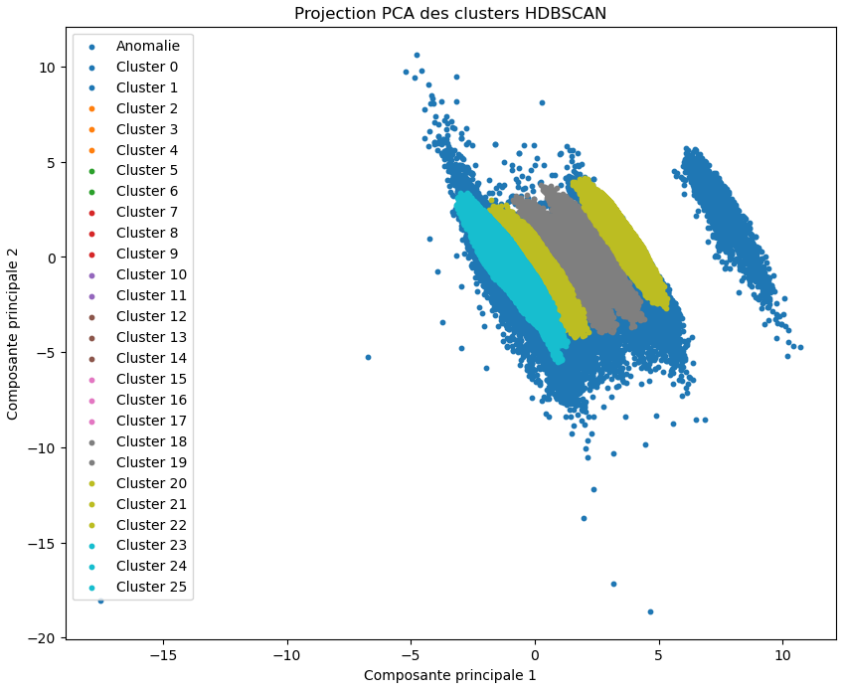


FIGURE 36 – Visualisation des clusters HDBSCAN et des points considérés comme bruit (gris)

Comparatif des méthodes

Méthode	Anomalies détectées	Observations normales
Isolation Forest	7 730	765 187
LOF	7 730	765 187
KMeans	402 155	19 2065
HDBSCAN	127 885 (bruit)	645 032

TABLE 5 – Comparaison des méthodes de détection d'anomalies

5.8 Réduction de dimension et visualisation des résultats

Afin de visualiser la répartition des anomalies et des clusters détectés, une réduction de dimension a été effectuée à l'aide de l'analyse en composantes principales (PCA). Cette étape permet de projeter les données multivariées sur un plan à deux dimensions tout en conservant au maximum la variance informative.

— **Objectif :** permettre une représentation visuelle des regroupements de données (clusters) et des points considérés comme anomalies.

— **Données utilisées :** toutes les variables numériques standardisées après encodage.

— **Méthode :** `PCA` de scikit-learn, avec visualisation des deux premières composantes principales.

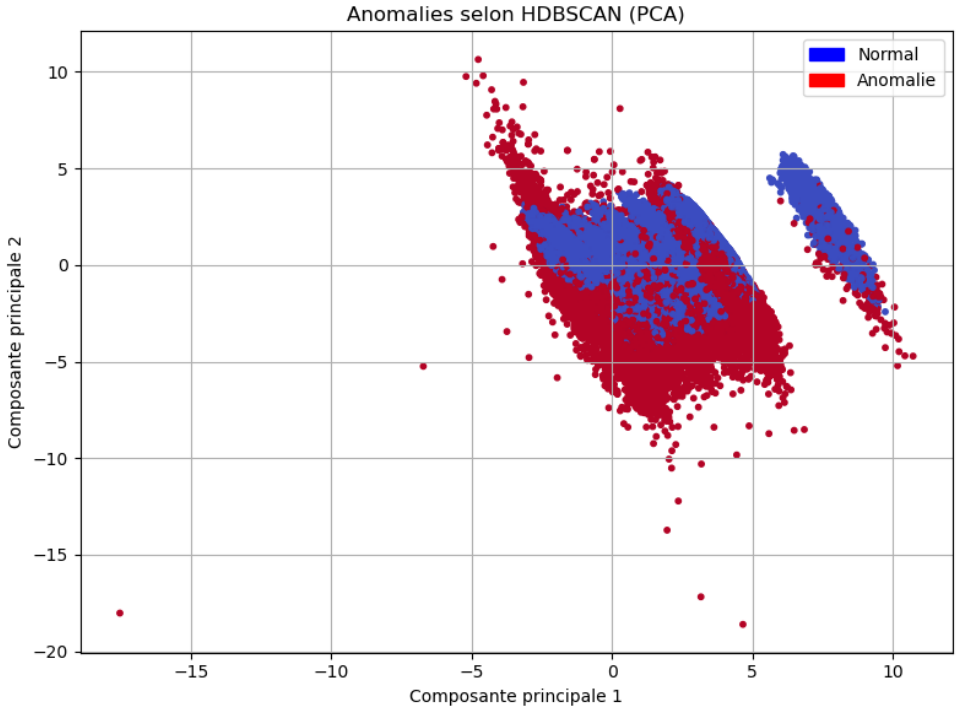


FIGURE 37 – Projection PCA avec surlignage des clusters HDBSCAN

Sur cette projection, on distingue clairement :

— Les groupes compacts correspondant aux clusters identifiés par HDBSCAN.

— Un nuage diffus de points écartés : ces derniers sont les observations marquées comme **bruit**, considérées comme anomalies.

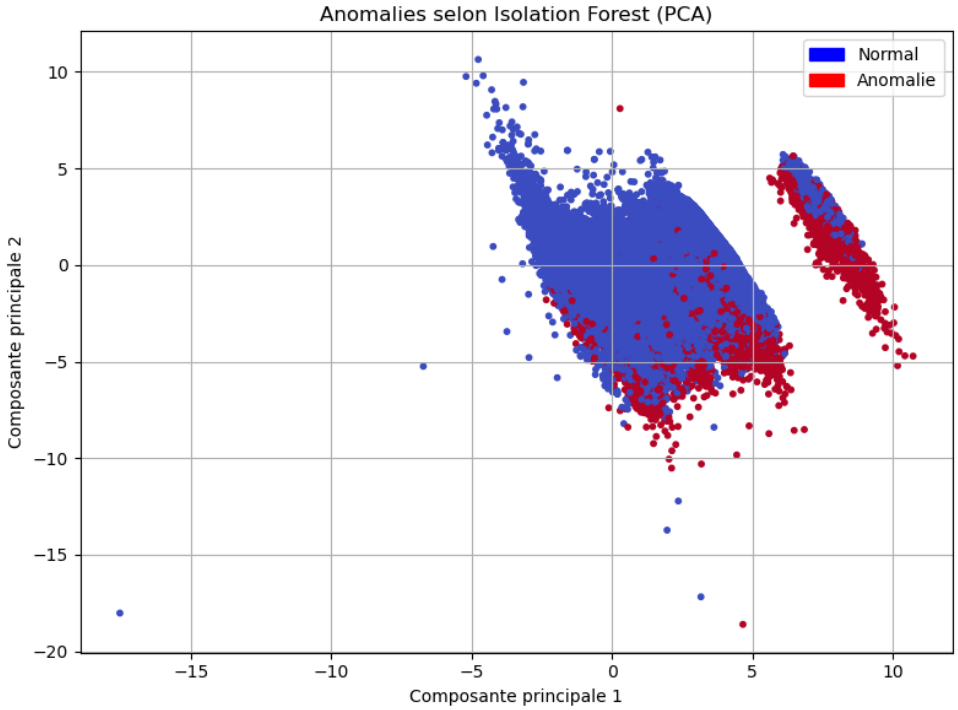


FIGURE 38 – Projection PCA des anomalies détectées par Isolation Forest

Les anomalies selon Isolation Forest apparaissent dispersées dans l'espace projeté, souvent en périphérie des zones denses, confirmant leur caractère inhabituel.

5.9 Analyse croisée et complémentarité des approches

Chaque modèle présente des sensibilités différentes :

— `HDBSCAN` est très efficace pour identifier des groupes cohérents et en rejeter les cas isolés.

— `Isolation Forest` est plus généraliste, détectant des cas atypiques mais bien répartis.

— `LOF` est adapté aux micro-anomalies dans des zones très denses.

— `KMeans` sert davantage d'outil de comparaison qu'un détecteur formel.

Ces approches peuvent donc être vues comme **complémentaires** plutôt que concurrentes. L'utilisation combinée de ces méthodes renforce la robustesse de l'analyse, en évitant de reposer sur une seule hypothèse d'anomalie.

5.10 Conclusion de chapitre

Ce projet a démontré la pertinence de la détection d'anomalies dans le contexte de la sécurité routière. En exploitant des données météorologiques, temporelles et géographiques, il a été possible de :

— Nettoyer et transformer efficacement un grand jeu de données,

— Mettre en place plusieurs modèles de détection d'anomalies,

— Visualiser et interpréter les résultats de façon intuitive,

— Justifier le choix de `HDBSCAN` pour sa capacité à isoler le bruit de manière naturelle,

— Comparer les comportements de chaque algorithme selon des critères concrets.

Ces travaux posent les bases pour des applications futures en temps réel ou pour le déclenchement automatique d'alertes lors d'événements anormaux dans les réseaux de transport. L'approche pourrait également être enrichie par l'ajout de variables supplémentaires (trafic, type de route, etc.) ou via des méthodes plus avancées telles que les autoencodeurs profonds.

6 Conclusion

Ce projet avait pour objectif d'analyser en profondeur un vaste jeu de données d'accidents de la route aux États-Unis afin d'en tirer des enseignements utiles à la compréhension, la classification et la détection de situations atypiques. Pour cela, plusieurs axes complémentaires ont été explorés.

Dans un premier temps, un travail de prétraitement a été réalisé. Ce nettoyage des données brutes, incluant la gestion des valeurs manquantes, la standardisation des variables et leur encodage, a permis de faciliter l'ensemble des analyses.

La prédiction de la sévérité des accidents a été abordée via plusieurs modèles de classification. Le modèle Gradient Boosting optimisé s'est démarqué avec des performances élevées. Des techniques de rééquilibrage des classes ont été mises en place afin de corriger les biais liés à la distribution initialement très déséquilibrée.

La segmentation non supervisée a permis de faire apparaître différents profils d'accidents à partir de critères météorologiques, temporels ou structurels. Cette approche a révélé des regroupements cohérents et a montré que certaines conditions sont plus fréquemment associées à des accidents graves, comme une visibilité réduite ou la nuit.

Enfin, une détection d'anomalies multivariée a été conduite à l'aide de plusieurs algorithmes : Isolation Forest, LOF, HDBSCAN et KMeans. L'analyse croisée de ces méthodes a permis d'identifier des événements singuliers pouvant constituer des cas critiques pour la sécurité routière. HDBSCAN s'est avéré très adapté dans ce contexte, notamment grâce à sa capacité à détecter des clusters de densité variable et à isoler naturellement les points de bruit.