Projet Mutualisé - Analyse et prédiction des accidents de la route

Groupe Vinland:
Sofian Braighith
Alicia Houset
Clément Martin
Djibril Traore

20 juin 2025

Table des matières

1 Introduction **3**

2 Traitement des données **4**
2.1 Chargement et inspection initiale 4
2.2 Nettoyage des données 4
2.3 Création de nouvelles variables 5
2.4 Encodage des variables 5

3 Prédiction de la sévérité d'un accident **6**
3.1 Objectif 6
3.2 Approche 6
3.3 Modèles testés 6
3.4 Limites et pistes d'amélioration 7

4 Prédiction de la distance **8**
4.1 Approche 8
4.2 Prétraitement 8
4.3 Entraînement de modèles 9
4.4 Fine Tuning 9
4.5 Analyse de l'importance des features 10

5 Prédiction des valeurs manquantes **11**
5.1 Choix de la colonne à prédire 11
5.2 Prétraitement 11
5.3 Entrainmenet de modèles 12
5.4 Génération de valeurs manquantes 12

6 Evaluation pratique des données générées **13**
6.1 Nouveau prétraitement 13
6.2 Résultats de la prédiction de la sévérité 13
6.3 Résultats de la prédiction de la distance 14
6.3.1 Conclusion sur l'apport des données générées 14

7 Perspectives et améliorations **16**

8 Conclusion **17**

1 Introduction

Les accidents de la route ont un impact considérable sur la société, que ce soit en termes de pertes humaines, de blessures ou de conséquences économiques. Chaque année, des milliers d'incidents sont recensés, et mieux les comprendre peut jouer un rôle clé dans la réduction des risques et l'amélioration des politiques de sécurité routière.

L'analyse des données liées aux accidents permet d'identifier des facteurs récurrents, de mettre en évidence des zones ou périodes à risque, et d'anticiper les situations potentiellement graves. Dans ce projet, nous nous sommes intéressés à l'exploitation de données à grande échelle afin de modéliser certains phénomènes liés aux accidents.

Notre premier objectif a été de prédire la sévérité d'un accident en fonction de différentes informations contextuelles comme la localisation, la météo, l'heure ou le type de route. Ce type de prédiction peut avoir des applications concrètes, par exemple dans l'optimisation des interventions de secours ou la prévention des accidents.

Le second objectif a été de prédire la distance couverte lors d'un accident en fonction des mêmes informations que pour la tâche précédente.

Enfin, nous avons mené d'autres analyses pour améliorer la qualité des données et enrichir la compréhension du phénomène. Pour ce faire, le dernier objectif consistait à prédire les valeurs manquantes. Cette tâche renforce la solidité des modèles construits et ouvre la voie à d'autres types d'analyses.

2 Traitement des données

2.1 Chargement et inspection initiale

Pour ce projet, nous avons utilisé le jeu de données *US Accidents* (version de mars 2023), disponible publiquement sur la plateforme Kaggle. Ce jeu de données contient des informations détaillées sur plus de 7,7 millions d'accidents survenus aux États-Unis entre 2016 et 2023. Il regroupe des données temporelles, géographiques, météorologiques ainsi que des indications sur les conditions de circulation.

Compte tenu de sa taille et de sa structure, le traitement de ce dataset a nécessité l'utilisation d'un environnement adapté au traitement de données volumineuses. Dans notre cas nous avons fait l'integralite de ce projet avec **Apache Spark** et plus precisemetn **PySpark**.

Une première phase d'inspection a permis d'identifier les variables disponibles, de repérer les colonnes contenant un grand nombre de valeurs manquantes et d'évaluer la distribution des valeurs dans les colonnes clés, comme la gravité des accidents ou leur répartition temporelle.

Cette étape exploratoire a également mis en évidence une forte hétérogénéité dans la qualité des données selon les États, les périodes ou les types d'accidents, ce qui a motivé certaines des décisions prises lors des étapes suivantes du nettoyage et de la préparation des données.

2.2 Nettoyage des données

Une fois le jeu de données chargé et exploré, nous avons réalisé un nettoyage afin de le rendre plus cohérent et utilisable pour l'analyse.

Dans un premier temps, nous avons supprimé les colonnes contenant un grand nombre de valeurs manquantes. Plus précisément, toutes les colonnes avec plus de 100 000 valeurs absentes ont été retirées, car elles apportaient trop peu d'informations exploitables. Cela concerne, par exemple, des variables liées à la météo ou à la localisation de fin, comme `End_Lat`, `End_Lng`, `Weather_Timestamp`, `Temperature(F)`, `Humidity(%)`, `Wind_Direction` ou encore `Precipitation(in)`.

Nous avons ensuite retiré certaines colonnes jugées non pertinentes pour la suite du projet. C'est le cas de `ID`, `Source` et `Description`, qui n'apportent pas d'information utile pour la modélisation.

Pour les colonnes de type catégoriel contenant encore quelques valeurs manquantes, nous avons remplacé celles-ci par la valeur la plus fréquente de chaque colonne. Cela permet de garder un maximum de données tout en assurant une certaine cohérence.

Enfin, nous avons également supprimé plusieurs colonnes à très forte diversité (comme `Street`, `City`, `County`, `Zipcode`, `Country` et `Airport_Code`) afin de réduire la complexité du jeu de données et limiter le bruit dans l'apprentissage automatique.

À l'issue de ces étapes, nous avons obtenu un jeu de données plus simple, plus propre, et mieux adapté aux traitements à venir.

2.3 Création de nouvelles variables

Pour enrichir le jeu de données initial et améliorer la qualité des prédictions, nous avons créé de nouvelles variables à partir de celles déjà présentes.

Nous avons tout d'abord extrait des informations temporelles à partir des colonnes `Start_Time` et `End_Time`. Ces horodatages ont été décomposés en sous-composantes telles que l'année, le mois, le jour, l'heure, la minute et la seconde. Ces variables permettent de mettre en évidence des effets liés au calendrier ou au moment de la journée, qui peuvent avoir un impact sur la gravité des accidents.

Nous avons ensuite calculé une nouvelle variable appelée `duration`, qui représente la durée de l'accident en minutes. Elle est obtenue en calculant la différence entre les heures de début et de fin. Cette information peut refléter indirectement la sévérité ou la complexité d'un accident.

Enfin, deux variables indicatrices ont été ajoutées :
— `is_weekend` : vaut 1 si l'accident s'est produit un samedi ou un dimanche, 0 sinon.
— `is_night` : vaut 1 si l'accident a eu lieu avant 6 heures du matin, 0 sinon.
Ces variables permettent d'intégrer des dimensions temporelles souvent importantes dans l'analyse des comportements routiers. Elles contribuent également à rendre les modèles plus sensibles au contexte dans lequel surviennent les accidents.

2.4 Encodage des variables

Avant de pouvoir entraîner les modèles de machine learning, certaines colonnes du jeu de données ont été transformées afin d'être compatibles avec les algorithmes utilisés.

Tout d'abord, les colonnes de type booléen, représentant des informations binaires (par exemple `Amenity`, `Bump`, `Railway`, etc.), ont été converties en valeurs numériques 0 ou 1. Cette conversion permet aux modèles de les interpréter correctement sans perte d'information.

Ensuite, les colonnes contenant des données textuelles ont été encodées avec l'outil `StringIndexer`, qui attribue un identifiant numérique à chaque modalité. Cela a été appliqué, par exemple, aux colonnes `Sunrise_Sunset`, `Civil_Twilight`, `Timezone` ou encore `State`. Cet encodage permet d'utiliser des variables catégorielles dans les modèles sans devoir manipuler directement du texte.

Enfin, toutes les variables numériques (y compris celles dérivées ou encodées) ont été regroupées dans un vecteur unique à l'aide de `VectorAssembler`. Cette étape est indispensable pour fournir aux modèles un format d'entrée structuré et standardisé.
Ces transformations ont permis d'obtenir un jeu de données prêt pour l'apprentissage automatique, avec des variables toutes numériques et exploitables par les algorithmes de classification.

3 Prédiction de la sévérité d'un accident

3.1 Objectif

L'objectif principal de ce projet est de prédire le niveau de gravité d'un accident de la route à partir des informations disponibles avant ou au moment de l'événement (heure, localisation, météo, type de route, etc.). Le niveau de gravité est défini par la variable `Severity`, qui prend des valeurs entières de 1 (accident mineur) à 4 (accident très grave).

3.2 Approche

Nous avons formulé ce problème comme une tâche de classification supervisée à quatre classes. Le modèle apprend à prédire la sévérité d'un accident à partir de ses caractéristiques, en se basant sur des exemples passés. Le jeu de données a été séparé en deux parties : un ensemble d'entraînement et un ensemble de test, afin d'évaluer objectivement les performances des modèles.

3.3 Modèles testés

Nous avons testé plusieurs modèles de classification : un modèle Random Forest, une régression logistique et un réseau de neurones (Multilayer Perceptron). Tous les modèles ont été évalués selon quatre métriques : précision globale (*accuracy*), précision pondérée, rappel (*recall*) et F1-score. Ces métriques permettent de mieux évaluer les performances sur des classes déséquilibrées.

Random Forest Une première version du modèle Random Forest a été entraînée avec `numTrees = 50`, `maxDepth = 5` et `maxBins = 64`, ce qui a donné une précision de 79,72 % et un F1-score de 70,73 %. Toutefois, le modèle prédisait très majoritairement la classe 2, qui représente environ 80 % des accidents.

Une seconde version, optimisée avec `numTrees = 75`, `maxDepth = 10` et `maxBins = 128`, a permis d'améliorer les performances générales : précision de 81,62 % et F1-score de 76,03 %. Cependant, les classes 1, 3 et 4 restent sous-représentées dans les prédictions. Par exemple, pour cette version, la classe 2 obtient un F1-score de 0,8962, tandis que les classes 1 et 4 plafonnent à respectivement 0,0301 et 0,0290.

Régression logistique Ce modèle linéaire a été utilisé avec une régularisation mixte L1/L2 (`elasticNetParam = 0.8`). Il a obtenu une précision de 79,26 % et un F1-score de 71,74 %. Une recherche d'hyperparamètres (grid search) a permis d'obtenir un modèle légèrement meilleur avec `regParam = 0.1` et `elasticNetParam = 0.25`, atteignant une précision de 79,76 % et un F1-score de 70,78 %.

Multilayer Perceptron (MLP) Un réseau de neurones à trois couches (128, 64, 32 neurones) a également été testé. Il atteint une précision de 79,72 % et un F1-score de 70,73 %. Ces résultats sont comparables à ceux du Random Forest non optimisé, mais le temps d'entraînement est plus élevé, et le modèle est plus difficile à interpréter.

Comparaison des performances Le tableau suivant résume les principales métriques pour les modèles testés :

Modèle	Accuracy	Precision	Recall	F1-Score
Random Forest (v1)	0.7972	0.6355	0.7972	0.7073
Random Forest (optimisé)	0.8162	0.8030	0.8162	0.7603
Régression logistique	0.7926	0.7040	0.7926	0.7174
Régression logistique (opt)	0.7976	0.6362	0.7976	0.7078
Multilayer Perceptron	0.7972	0.6355	0.7972	0.7073

Tous les modèles obtiennent des scores de précision relativement proches, mais montrent une forte tendance à prédire la classe majoritaire. Les classes 1, 3 et 4, bien que représentatives d'accidents plus graves ou plus rares, sont souvent mal identifiées.

3.4 Limites et pistes d'amélioration

Malgré des performances globales satisfaisantes, tous les modèles présentent des difficultés à bien prédire les classes minoritaires (1, 3 et 4), ce qui est lié à leur faible proportion dans le jeu de données. Par exemple, la classe 2 représente plus de 80 % des cas, ce qui pousse les modèles à la prédire par défaut. Cela se reflète dans les F1-scores par classe : pour le modèle Random Forest optimisé, la classe 2 obtient un F1-score de 0.89, contre seulement 0.03 pour la classe 1 et 0.03 pour la classe 4.

Pour mieux équilibrer les prédictions, plusieurs pistes pourraient être envisagées : ajustement des poids de classes, suréchantillonnage des classes rares, ou encore sélection de métriques adaptées comme le F1-score macro. Une exploration plus poussée de la matrice de confusion et des métriques par classe pourrait guider l'amélioration des performances.

Enfin, certaines limitations techniques ont été rencontrées, notamment une erreur de mémoire lors de l'entraînement de modèles Random Forest plus complexes, ce qui a limité les possibilités d'optimisation.

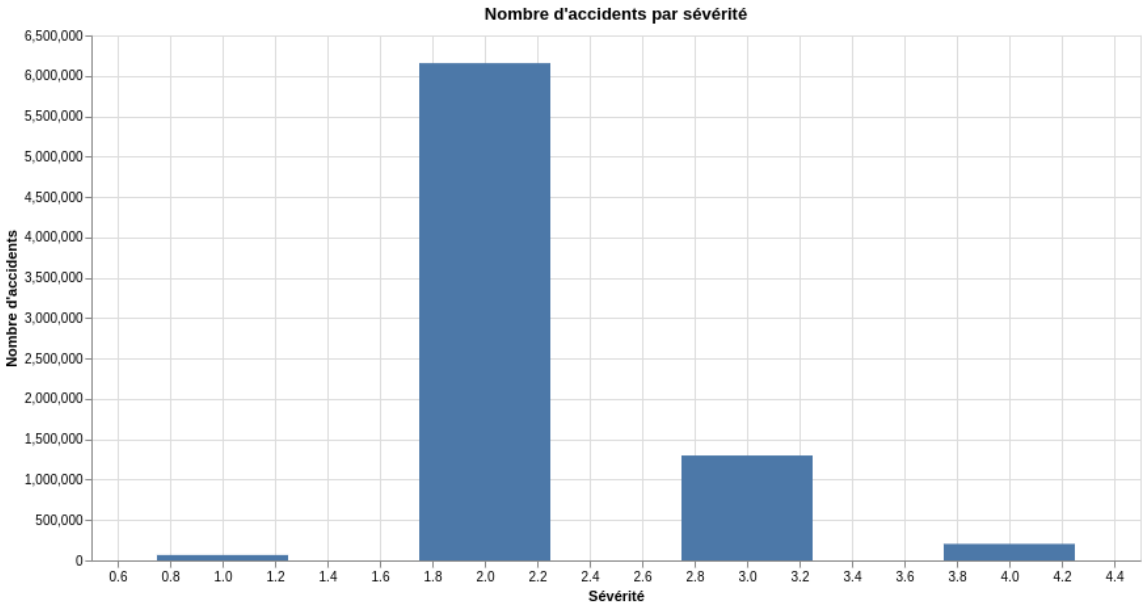

FIGURE 1 – Distribution de la sévérité

4 Prédiction de la distance

Parmi les informations intéressantes du dataset, il y avait la colonne ***Distance(mi)***. Celle-ci contient la distance en miles sur laquelle le trafic a été affecté par l'accident. L'objectif était donc de prédire sur quelle distance le trafic est affecté suite à un accident. Cette tâche n'est pas triviale, car l'impact sur le trafic peut dépendre d'un nombre énorme de facteurs externes dont nous ne disposons pas forcément dans le dataset.

4.1 Approche

Le choix d'utiliser des algorithmes de régression s'impose naturellement, car notre variable cible représente la distance d'impact sur le trafic, soit une grandeur quantitative continue pouvant prendre n'importe quelle valeur réelle positive dans l'intervalle observé.

4.2 Prétraitement

La variable cible ***Distance(mi)*** a nécessité un prétraitement spécifique pour optimiser les performances des modèles de régression. À la suite du prétraitement présenté dans la partie 2, nous avons converti cette variable de miles en mètres en appliquant le facteur de conversion standard ($1mi = 1609.344m$), permettant ainsi d'obtenir une unité de mesure mieux répartie. En effet, sans ça, la majorité des distances était comprise entre 0 et 2 miles. Cette conversion n'a cependant pas arrangé la distribution fortement asymétrique des distances d'impact, avec une concentration importante des valeurs entre 161 et 3218 mètres (après conversion), une moyenne largement supérieure à la médiane et la présence de valeurs extrêmes supérieures à 16 kilomètres.

Face à cette distribution asymétrique, nous avons dans un premier temps supprimé les lignes avec des distances égales à 0 *(pratiquement 50% du dataset)* ainsi que les valeurs aberrantes supérieures à 10 miles *(16093.44 mètres)*. Ensuite, nous avons appliqué une transformation logarithmique selon la formule. $Distance\_Log = log(Distance(m) + 1)$. Comme on peut le voir sur les graphiques qui suivent, cette normalisation a permis de réduire significativement l'asymétrie de la distribution.

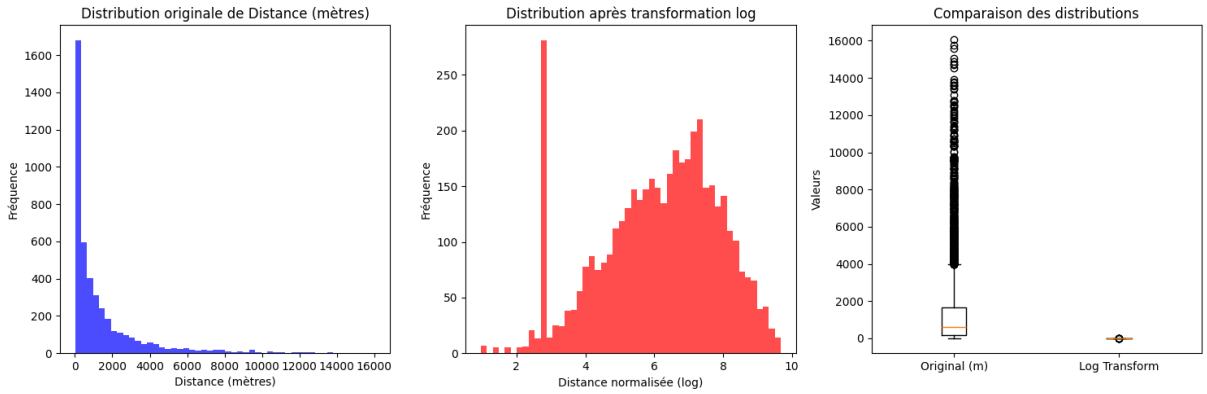

FIGURE 2 – Comparaison des distributions avant et après la normalisation logarithmique

4.3 Entraînement de modèles

Pour cette problématique, nous avons entraîné 4 modèles différents, tous avec une seed de 42 :

Linear Regression :
- maxIter : 100
- regParam : 0.01
- elasticNetParam : 0.1

Random Forest :
- numTrees : 100
- maxDepth : 15
- maxBins : 64

Gradient Boosted Tree Regressor :
- maxIter : 100
- maxDepth : 10
- maxBins : 64

Decision Tree Regressor :
- maxDepth : 15
- minInstancesPerNode : 1
- maxBins : 64

Pour évaluer la performance de nos modèles de régression, nous utilisons trois métriques complémentaires :
- **RMSE** (Root Mean Square Error) : pénalise fortement les erreurs importantes, s'exprime ici en mètres.
- **MAE** (Mean Absolute Error) : moins sensible aux valeurs aberrantes que le RMSE, s'exprime ici en mètres.
- $\mathbf{R^2}$ (coefficient de détermination) : varie de 0 à 1 où une valeur proche de 1 signifie une excellente capacité prédictive.

Voici les résultats des différents modèles pour chaque métrique :

Modèle	RMSE	MAE	R^2	Temps d'entraînement (s)
GBT	1.1282	0.8416	0.5709	704.1507
Random Forest	1.2009	0.9076	0.5138	708.5547
Decision Tree	1.2321	0.9177	0.4882	35.2293
Linear Regression	1.5625	1.2817	0.1769	18.9331

TABLE 1 – Classement des modèles par performance décroissante en R^2

4.4 Fine Tuning

Après avoir identifié le modèle GBT comme le plus performant avec un R^2 de 0.57, nous avons testé différentes configurations pour améliorer ses résultats. Six variantes ont été évaluées en changeant les paramètres principaux :
- maxIter : nombre d'arbres (100 à 120)
- maxDepth : profondeur des arbres (10 à 12 niveaux)
- stepSize : vitesse d'apprentissage (0.06 à 0.12)
- maxBins : nombre de divisions possibles (64 à 128).
- seed : 42

Cette approche simple et directe permet de trouver la meilleure combinaison de paramètres parmi celles proposées et d'identifier la configuration qui donne le meilleur score R^2 pour prédire avec précision la distance d'impact des accidents sur le trafic. Finalement notre meilleur model est un GBT avec :

- maxIter : 120
- maxDepth : 12
- stepSize : 0.06
- maxBins : 128

Voici ses resultat finaux :

Modèle	**RMSE**	**MAE**	**R^2**
Best GBT	1.1054	0.8201	0.5880

TABLE 2 – Meilleur modèle GBT en R^2

4.5 Analyse de l'importance des features

L'analyse de l'importance des features permet d'identifier quelles caractéristiques d'un accident ont le plus d'influence sur la distance d'impact trafic. Grâce au modèle Random Forest, nous pouvons calculer un score d'importance pour chaque variable utilisée dans la prédiction, ce qui nous indique leur contribution relative au résultat final. Voici une representation graphique des features les plus importantes pour le random forest.

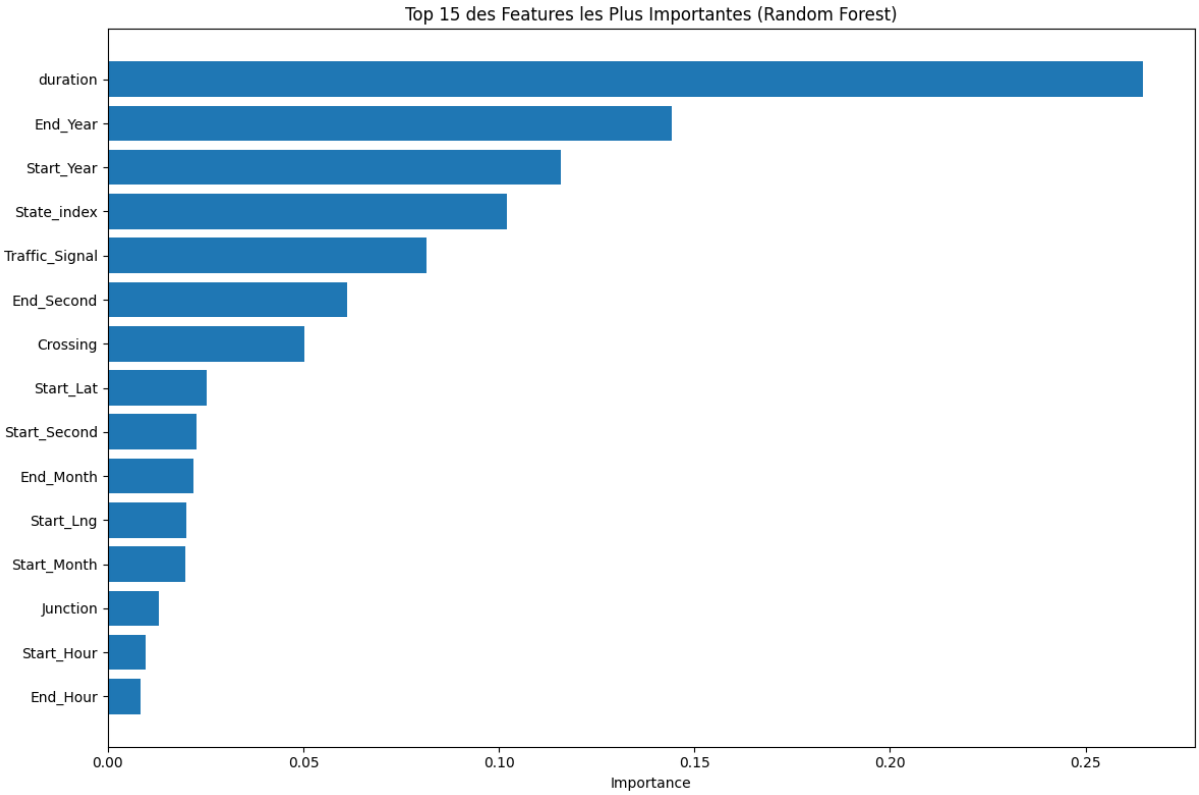

FIGURE 3 – Classement des features les plus importantes pour le random forest.

Les informations fournies par le Random Forest nous permettent aussi de dire que les 10 meilleures features expliquent 89.0% de la variance.

5 Prédiction des valeurs manquantes

Le jeu de données utilisé contient un grand nombre de valeurs manquantes, dans plusieurs colonnes. Pour les autres tâches, nous avons géré cela en retirant les colonnes ou lignes, ou en faisant de l'imputation de valeurs. Mais retirer des colonnes entières du jeu de données peut faire perdre des informations importantes à un modèle. Et les methodes d'imputation simples ne sont pas très efficaces, surtout lorsque la proportion de valeurs manquantes est large. Nous avons donc essayer d'entrainer un modèle à prédire des valeurs manquantes.

5.1 Choix de la colonne à prédire

Afin de choisir quelle colonne serait la plus intéréssante à prédire, nous avons regardé les colonnes avec le plus de valeurs manquantes :

— End_Lng - 3 402 762
— End_Lat - 3 402 762
— Precipitation(in) - 2 203 586
— Wind_Chill(F) - 1 999 019
— Wind_Speed(mph) - 571 233

Après une analyse de la difference entre la position de départ et la position de fin pour les lignes contenant les deux informations, nous avons conclut que les valeurs étaient très proches et donc que la position de fin n'apportait pas des informations très intéréssantes. Nous avons donc retiré les colonnes End_Lng et End_Lat.

Nous avons ensuite fait une analyse des correlations des 3 colonnes suivantes avec les autres colonnes contenant des types numeriques. Nous avons trouvé que la colonne Wind_Chill a une correlation de 1 avec la temperature, donc guarder les deux est inutile.

Il nous restais donc la précipitation et la vitesse du vent. Étant donné que la colonne des précipitation contenait 4 fois plus de valeurs manquantes, nous avons décidé de prédire celle-ci.

5.2 Prétraitement

Nous avons retiré les colonens précedemment identifiés comme redondantes ou pas intéréssantes pour le modèle. Les colonnes contenant du texte (à part Description et celles contenant trop de valeurs différentes) ont été transformé en vecteur binaires (bag-of-words). Nous avons extrait à nouveau des colonnes temporelles à partir des dates que nous avons ensuite retiré. Pour finir, nous avons appliqué une normalisation au colonnes contenant des valeurs numériques.

5.3 Entrainmenet de modèles

Nous avons entrainé 4 modèles :
— Linear Regression
— Decision Tree
— Random Forest
— Gradient Boosted Trees

Comme nous pouvons le voir dans la table de comparaison 3, le modèle avec la meilleure performance est le Gradient Boosted Trees. Nous avons évalué la qualité de l'imputation par la moyenne à titre de comparaison, battu par tout les modèles.

Model	**RMSE**	**R^2**
Gradient Boosted	0.066077	0.3950033
Decision Tree	0.076124	0.1970305
Random Forest	0.078877	0.1379123
Linear Regression	0.080844	0.09436297
Dummy (Mean)	0.084952	-3.246162e-07

TABLE 3 – Comparaison des modèles

5.4 Génération de valeurs manquantes

Nous avons donc utilisé le modèle Gradient Boosted Trees pour générer les valeurs manquantes. Ayant retiré les lignes contenant des valeurs manquantes dans le prétraitement, nous n'avons pas pu générer des valeurs pour toutes les lignes, ayant au final pu completer 3/4 des lignes sur environ 2 million.

6 Evaluation pratique des données générées

Après avoir développé des modèles de prédiction de la sévérité des accidents de la route ainsi que de la distance de route impactée, puis mis en place une méthode pour générer les données manquantes, il est essentiel d'évaluer l'impact de ces données générées sur les performances de nos modèles. Cette section a pour objectif d'analyser les résultats obtenus lorsque ces modèles sont appliqués aux données complétées. Nous chercherons notamment à déterminer si les prédictions restent fiables et cohérentes, et dans quelle mesure les données synthétiques influencent la qualité des résultats par rapport aux données originales.

6.1 Nouveau prétraitement

Afin d'intégrer les valeurs générées dans notre jeu de données, un nouveau prétraitement a été nécessaire. Malgré l'étape de génération, certaines colonnes présentaient encore un nombre important de valeurs manquantes. C'est notamment le cas de la colonne Precipitation(in), qui compte encore plus de 500 000 valeurs manquantes.

Pour cette phase, nous avons donc choisi de conserver uniquement les colonnes contenant moins d'un million de valeurs manquantes. Comme précédemment, les valeurs manquantes restantes ont été remplies à l'aide de la valeur la plus fréquente (mode), une méthode simple mais efficace pour limiter la perte d'information. Enfin, toutes les lignes contenant encore des valeurs manquantes après cette étape ont été supprimées.

```
Nombre de lignes avant suppression des valeurs manquantes : 7728394

Nombre de lignes après suppresion des valeurs manquantes : 7081140
```

Ainsi, environ 700 000 lignes ont été retirées du jeu de données.

Une fois ce nettoyage effectué, nous avons procédé à l'encodage des nouvelles colonnes catégorielles afin de les rendre compatibles avec nos modèles de machine learning. Les colonnes temporelles ont également été transformées, en les splittant en composantes exploitables (jour, mois, année, heure, etc.) pour permettre une meilleure prise en compte des variations temporelles dans les prédictions.

6.2 Résultats de la prédiction de la sévérité

Comme mentionné précédemment, le modèle Random Forest s'était montré le plus performant lors de nos expérimentations initiales. Nous avons donc conservé ce modèle pour évaluer la prédiction de la sévérité des accidents sur les données enrichies.

Le meilleur modèle final retenu est un Random Forest configuré avec les hyperparamètres suivants :

- numTrees=25,
- maxDepth=10,
- maxBins=192,
- seed=42.

Les performances obtenues sont les suivantes :

```
=== Résumé des métriques Random Forest ===
Accuracy:  0.8235
Precision: 0.8113
Recall:    0.8235
F1-Score:  0.7685
```

Ces résultats montrent une amélioration globale des performances par rapport aux modèles précédemment évalués. L'intégration des données générées semble donc avoir eu un effet positif sur la capacité du modèle à prédire correctement la sévérité des accidents.

6.3 Résultats de la prédiction de la distance

Pour la tâche de régression visant à prédire la distance de route impactée par un accident, nous avons utilisé le modèle Gradient-Boosted Trees (GBT), qui avait montré de bonnes performances lors de nos tests initiaux.

Le modèle final a été entraîné avec les paramètres suivants :

- maxIter=120,
- maxDepth=12,
- stepSize=0.06,
- maxBins=192,
- seed=42.

Les résultats obtenus sont les suivants :

```
   === Résumé des performances GBT ===
RMSE (Root Mean Squared Error) : 1.11 mètres
MAE (Mean Absolute Error)      : 0.82 mètres
R² (coefficient de détermination) : 0.5778
Temps d'entraînement            : 8257.36 secondes
```

Ces résultats montrent que le modèle parvient à prédire la distance impactée avec une erreur relativement faible (moins de 1,2 mètre en moyenne). Toutefois, le coefficient de détermination ($R^2 = 0.5778$) est légèrement inférieur à celui obtenu précédemment ($R^2 = 0.5880$), ce qui indique une légère dégradation des performances. Cette baisse peut être liée à l'introduction des données générées, qui peuvent introduire un certain bruit dans les prédictions.

6.3.1 Conclusion sur l'apport des données générées

L'intégration de données générées a permis d'enrichir significativement notre jeu de données initial, en réduisant le volume de valeurs manquantes et en rendant exploitables des variables jusqu'alors incomplètes. Cet enrichissement a contribué à améliorer la couverture des cas d'étude, notamment en permettant d'entraîner les modèles sur un ensemble plus complet et plus homogène.

Cependant, les résultats obtenus montrent que l'apport des données générées n'est pas uniformément positif. Si, dans le cas de la prédiction de la sévérité des accidents, nous avons observé une amélioration des performances globales, ce n'est pas le cas pour la prédiction de la distance de route impactée, où une légère dégradation du coefficient de détermination (R^2) a été constatée.

Ces résultats suggèrent que, bien que la génération de données puisse être une solution efficace pour pallier les valeurs manquantes, elle peut également introduire un bruit ou une imprécision susceptible de pénaliser certains modèles, en particulier ceux sensibles aux valeurs exactes comme les modèles de régression.

En conclusion, la génération de données s'avère être une approche prometteuse, mais qui doit être utilisée avec discernement. Il est essentiel d'en évaluer l'impact selon les tâches de prédiction visées et de veiller à ce que les données synthétiques soient aussi représentatives et cohérentes que possible vis-à-vis des données réelles.

7 Perspectives et améliorations

Les résultats obtenus dans ce projet sont encourageants, mais plusieurs pistes peuvent être explorées pour améliorer encore les performances et la robustesse des modèles.

Tout d'abord une meilleure gestion du déséquilibre des classes. Dans la prédiction de la sévérité, les classes les plus graves (1, 3 et 4) sont largement sous-représentées. L'utilisation de techniques comme le suréchantillonnage des classes rares, la pondérati-classes,classes etc.

Ensuite, l'exploration de modèles avancés. Bien que les modèles utilisés (Random Forest, GBT, etc.) aient donné de bons résultats, d'autres approches comme les réseaux neuronaux.

De plus, l'amélioration de la génération de données. La génération de valeurs manquantes peut encore être optimisée. Nous pourrions aussi essayer de générées des données d'autres features et tester si cela nuît ou améliore nos modèles.

Enfin, le dataset de base contient beaucoup de features et par conséquent d'information. Cependant, très peu sont réellement utilisables (trop de valeurs manquantes, corrélation quasiment parfaite avec d'autres features, etc). En réalité, malgré la taille conséquente du dataset, il a été difficile de trouver des sujets intéressants ou tout simplement de le manipuler. Avoir un dataset plus complet avec, par exemple, des features sur la cause de l'accident, les blessures causées, ou tout simplement ajouter des cas où il n'y a pas d'accident pourrait améliorer tout ceci.

8 Conclusion

Ce projet montre qu'il est possible de prédire avec une bonne précision la sévérité des accidents de la route ainsi que la distance de route impactée, en s'appuyant sur un jeu de données riche et sur des modèles de machine learning bien calibrés. Les performances obtenues en classification comme en régression démontrent la pertinence de ces approches pour extraire des connaissances à partir de données complexes issues du monde réel.

L'ajout de données synthétiques pour combler les valeurs manquantes a constitué une étape clé de notre projet. Cette stratégie d'imputation a permis d'enrichir le jeu de données et de rendre exploitables certaines colonnes qui auraient autrement été écartées. Toutefois, nous avons observé que cette approche n'a pas le même impact selon les tâches de prédiction : elle s'est révélée bénéfique pour la classification de la sévérité des accidents, en améliorant la performance du modèle, mais légèrement défavorable pour la régression, possiblement en introduisant un léger bruit.

Malgré les limites du dataset, telles que des déséquilibres de classes, des valeurs manquantes ou encore une qualité inégale selon les features, nous sommes globalement satisfaits des résultats obtenus. Le projet met en lumière l'importance cruciale du prétraitement des données, du choix des modèles et de leur calibrage pour répondre efficacement à des problématiques concrètes de prédiction.

Enfin, ce travail ouvre la voie à plusieurs perspectives d'amélioration : affinement de l'imputation, tests de modèles plus avancés (réseaux de neurones, modèles hybrides), ou encore intégration de données contextuelles supplémentaires. Ces pistes pourraient permettre d'augmenter encore la précision des prédictions et la robustesse des solutions proposées dans le cadre d'une application réelle.