Building Claim Prediction

Ulysse Trin

2024-04-14

Introduction

Ce document vise à présenter la démarche analytique dans le cadre du Data Challenge ENS sur l'assurance des bâtiments. L'objectif est de prédire la probabilité qu'un bâtiment soit sujet à un sinistre durant une période donnée, en se basant sur ses caractéristiques. À travers ce challenge, nous aborderons la modélisation de la fréquence des sinistres, essentielle dans le processus de tarification en assurance non-vie.

Contexte

L'assurance bâtiment couvre les dommages susceptibles d'affecter la structure d'un logement (murs, toit, sols, et zones communes) causés par des événements comme les inondations, les tempêtes, les incendies ou le vandalisme.

Le défi proposé s'inscrit dans le processus de tarification de l'assurance non-vie, qui se décompose généralement en deux étapes :

- **Modélisation de la fréquence** : détermination du nombre attendu de sinistres qu'un assureur recevra pendant une période donnée.
- **Modélisation de la sévérité** : prédiction du coût moyen d'un sinistre.

Cependant, l'objectif de ce challenge est spécifique : il s'agit de prédire si un bâtiment déposera une réclamation d'assurance durant une période assurée, basé sur les caractéristiques du bâtiment. La variable cible est donc :

- 1 si le bâtiment a au moins un sinistre pendant la période assurée.
- 0 si le bâtiment n'a pas de sinistre pendant la période assurée.

Description des données

Les données en entrée contiennent les variables suivantes :

- **Identifiant** : variable permettant d'identifier le client dans la base de données de l'assureur. Variable d'identification.
- **EXPO** : temps assuré chez Generali sur l'année. Variable numérique.
- **superficief** : taille en m^2 du bâtiment assuré. Variable numérique.
- **Insee** : code géographique français pour localiser le bâtiment assuré. Variable géographique/catégorielle très utile pour ajouter des données externes.
- **target** : variable cible (0 : pas de réclamation, 1 : au moins une réclamation sur la période assurée).
- **Autres ft_i_categ** : caractéristiques anonymisées du bâtiment. Variables catégorielles.

Benchmark initial

Un benchmark rapide avec un modèle xgboost a été réalisé, atteignant un score NGC de 0.41. Il s'agit d'un modèle basique où toutes les variables catégorielles sont codées par étiquette. Les variables les plus importantes identifiées sont `superficief`, `ft_22_categ`, `EXPO`.

Métrique

La métrique utilisée pour ce challenge est le coefficient de Gini normalisé. L'objectif est de construire un modèle qui prédit l'ordre des sinistres. Un modèle est performant s'il détecte avec une plus grande probabilité les bâtiments ayant effectivement eu un sinistre.

Plan

Notre approche se déclinera en plusieurs étapes clés :

1. **Exploration et traitement des données** : Comprendre la distribution, la qualité, et les relations potentielles entre les variables.
2. **Modélisation** : Développement et entraînement de modèles prédictifs, en explorant différentes techniques et en optimisant leurs hyperparamètres.

- Arbre de classification
- XGBoost
- LGMBoost

3. **Évaluation et Sélection des Modèles** : Comparaison des modèles basée sur la métrique du coefficient de Gini normalisé et sélection du meilleur modèle.
4. **Conclusion**

Ce document détaillera chaque étape de notre démarche, justifiera nos choix méthodologiques et interprétera les résultats obtenus afin de fournir une compréhension approfondie du phénomène étudié.

1. Exploration et traitement des données

Pour commencer, nous vérifions si les packages nécessaires sont installés, et nous les installons s'ils ne le sont pas.

Importation des données : test (X_test), entraînement (X_train), cible (Y_train)

Lors de l'importation de X_test et X_train, nous procédons à la suppression de la première colonne. Cette colonne, souvent générée automatiquement lors de l'exportation des données depuis certaines applications ou bases de données, sert d'index. Cependant, dans le contexte de notre analyse, cette colonne d'index n'est pas nécessaire et peut même interférer avec notre traitement des données.

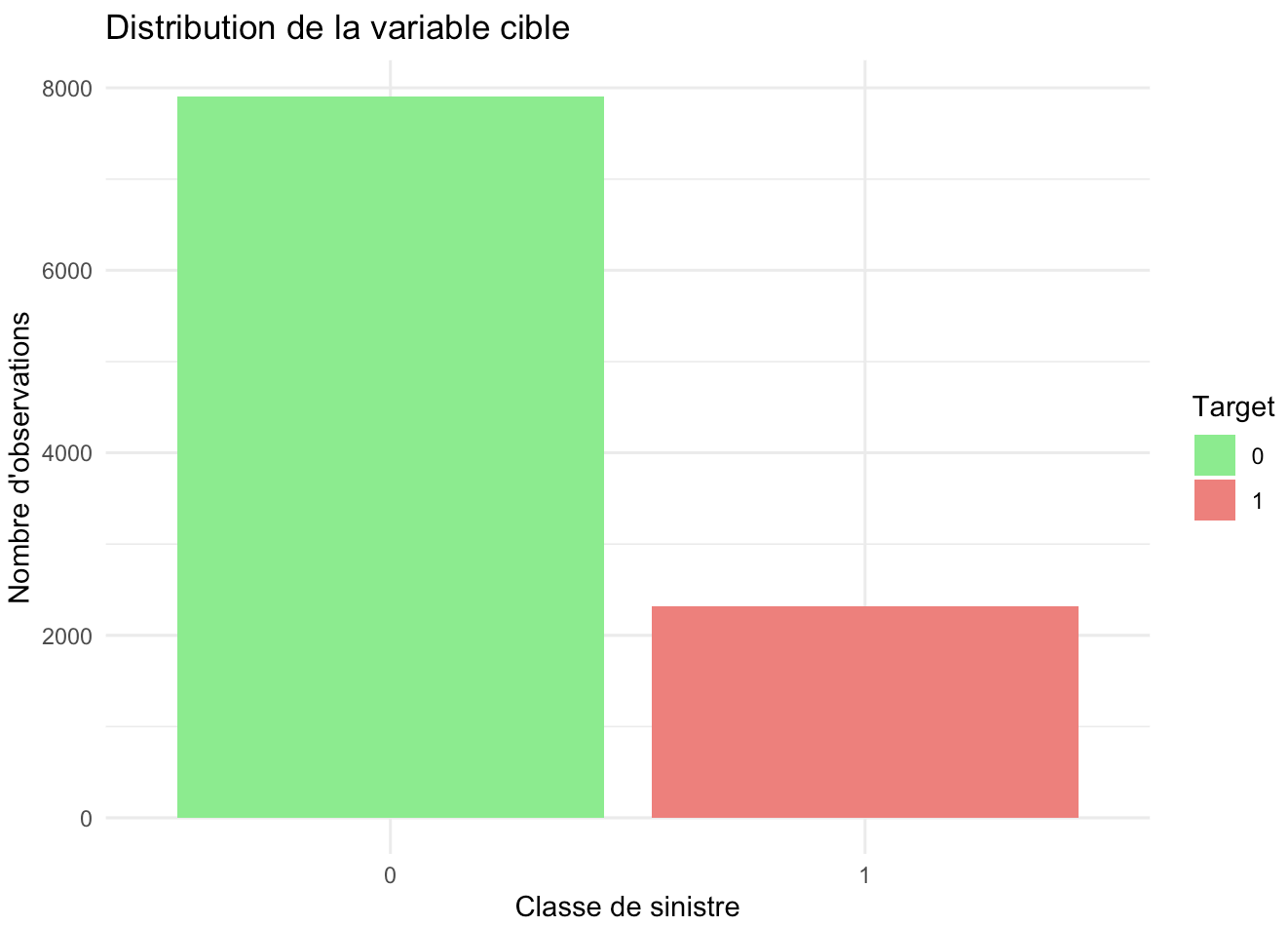
Exploration de la variable cible Y_train

```
##
##    0    1
## 7907 2322
```

La classe "0" apparaît 7907 fois et la classe "1" 2322 fois.

```
##
##        0        1
## 77.29983 22.70017
```

77.3% des données sont de la classe 0 et 22.7% de la classe 1, illustrant un déséquilibre où la majorité des observations appartient à la classe 0.



L'exploration de la variable cible Y_train met en évidence une distribution asymétrique entre les classes, ce qui soulève des considérations importantes pour la modélisation. Avec 7 907 observations appartenant à la classe 0 (pas de sinistre) et 2 322 à la classe 1 (au moins un sinistre), nous observons une répartition de 77.3% pour la classe 0 et de 22.7% pour la classe 1.

Cette distribution indique un déséquilibre significatif, où la majorité des observations ne signalent aucun sinistre. Cet état de fait peut engendrer des biais dans les modèles prédictifs, qui pourraient être enclins à favoriser la prédiction de la classe majoritaire. Cette prédominance pourrait diminuer la sensibilité du modèle aux cas réels de sinistres, affectant ainsi négativement la performance du modèle sur des données moins fréquentes mais critiques.

Importation de données de correspondance entre le code INSEE et le code postale

Nous importons les données de correspondance entre le code INSEE et le code postal afin de maximiser les possibilités d'enrichissement de notre modèle avec des données externes.

Enrichissement des données X_train et X_test avec les données de correspondance

Nous enrichissons les données d'entraînement (X_train) et de test (X_test) avec les données de correspondance en utilisant le code INSEE comme clé de jointure.

Vérification de l'enrichissement de X_test :

```
##  [1] "Identifiant"           "ft_2_categ"            "EXPO"
##  [4] "ft_4_categ"            "ft_5_categ"            "ft_6_categ"
##  [7] "ft_7_categ"            "ft_8_categ"            "ft_9_categ"
## [10] "ft_10_categ"           "ft_11_categ"           "ft_12_categ"
## [13] "ft_13_categ"           "ft_14_categ"           "ft_15_categ"
## [16] "ft_16_categ"           "ft_17_categ"           "ft_18_categ"
## [19] "ft_19_categ"           "superficief"           "ft_21_categ"
## [22] "ft_22_categ"           "ft_23_categ"           "ft_24_categ"
## [25] "Insee"                 "Nom_de_la_commune"     "Code_postal"
## [28] "Libellé_d_acheminement"
```

Vérification de l'enrichissement de X_train :

```
##  [1] "Identifiant"           "ft_2_categ"            "EXPO"
##  [4] "ft_4_categ"            "ft_5_categ"            "ft_6_categ"
##  [7] "ft_7_categ"            "ft_8_categ"            "ft_9_categ"
## [10] "ft_10_categ"           "ft_11_categ"           "ft_12_categ"
## [13] "ft_13_categ"           "ft_14_categ"           "ft_15_categ"
## [16] "ft_16_categ"           "ft_17_categ"           "ft_18_categ"
## [19] "ft_19_categ"           "superficief"           "ft_21_categ"
## [22] "ft_22_categ"           "ft_23_categ"           "ft_24_categ"
## [25] "Insee"                 "Nom_de_la_commune"     "Code_postal"
## [28] "Libellé_d_acheminement"
```

Intégration du numéro départemental

Nous ajoutons une nouvelle colonne intitulée "Numéro_Départements" créer à partir des deux premiers chiffres des codes postaux dans les ensembles de données X_train_enriched et X_test_enriched.

Importation de données sur le nom des départements et des régions

Enrichissement des données X_train et X_test avec le nom des départements et des régions

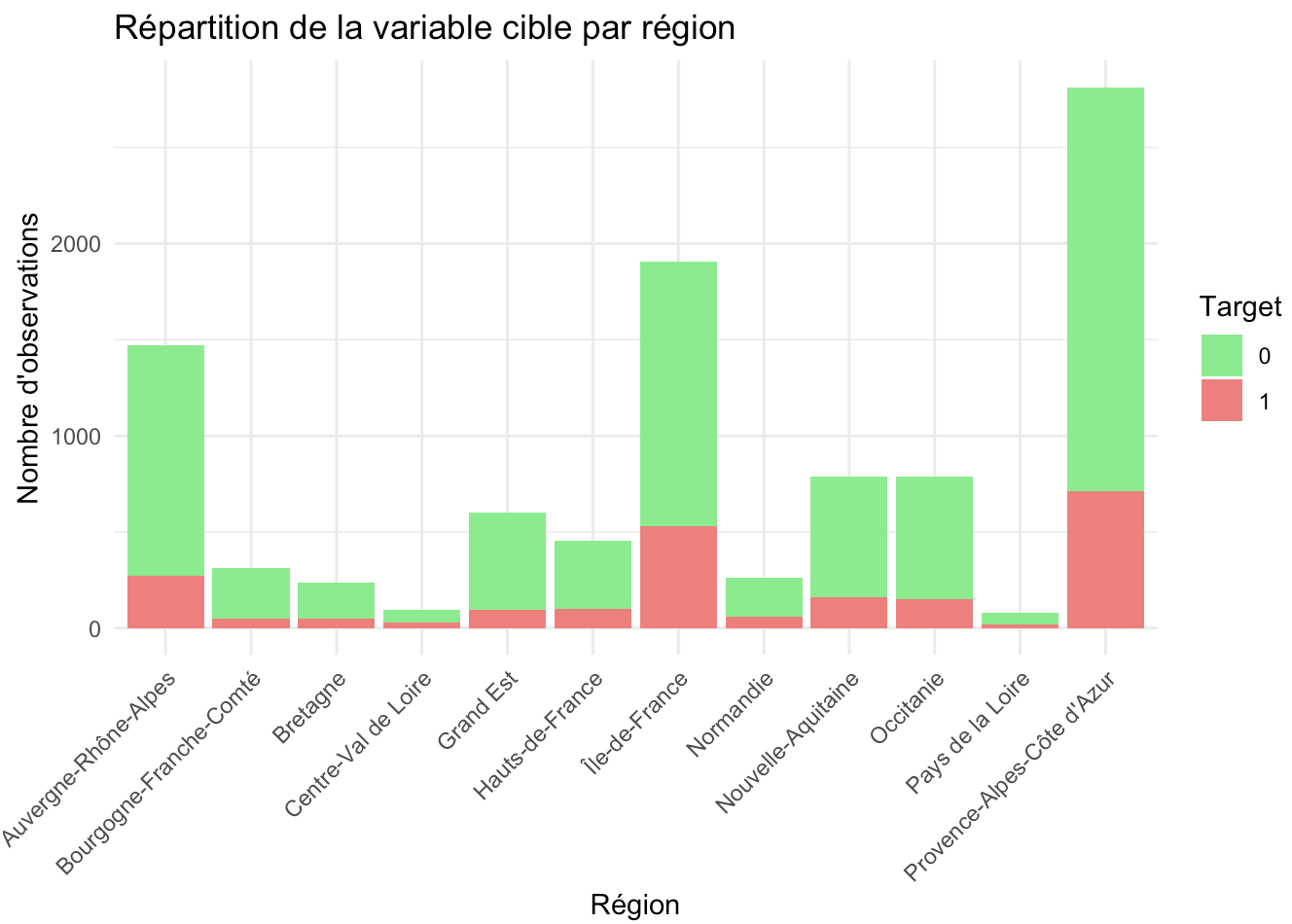
Nous enrichissons les ensembles de données X_train_enriched et X_test_enriched en ajoutant des informations sur les noms des départements et des régions.

Vérification de l'enrichissement de X_test_enriched :

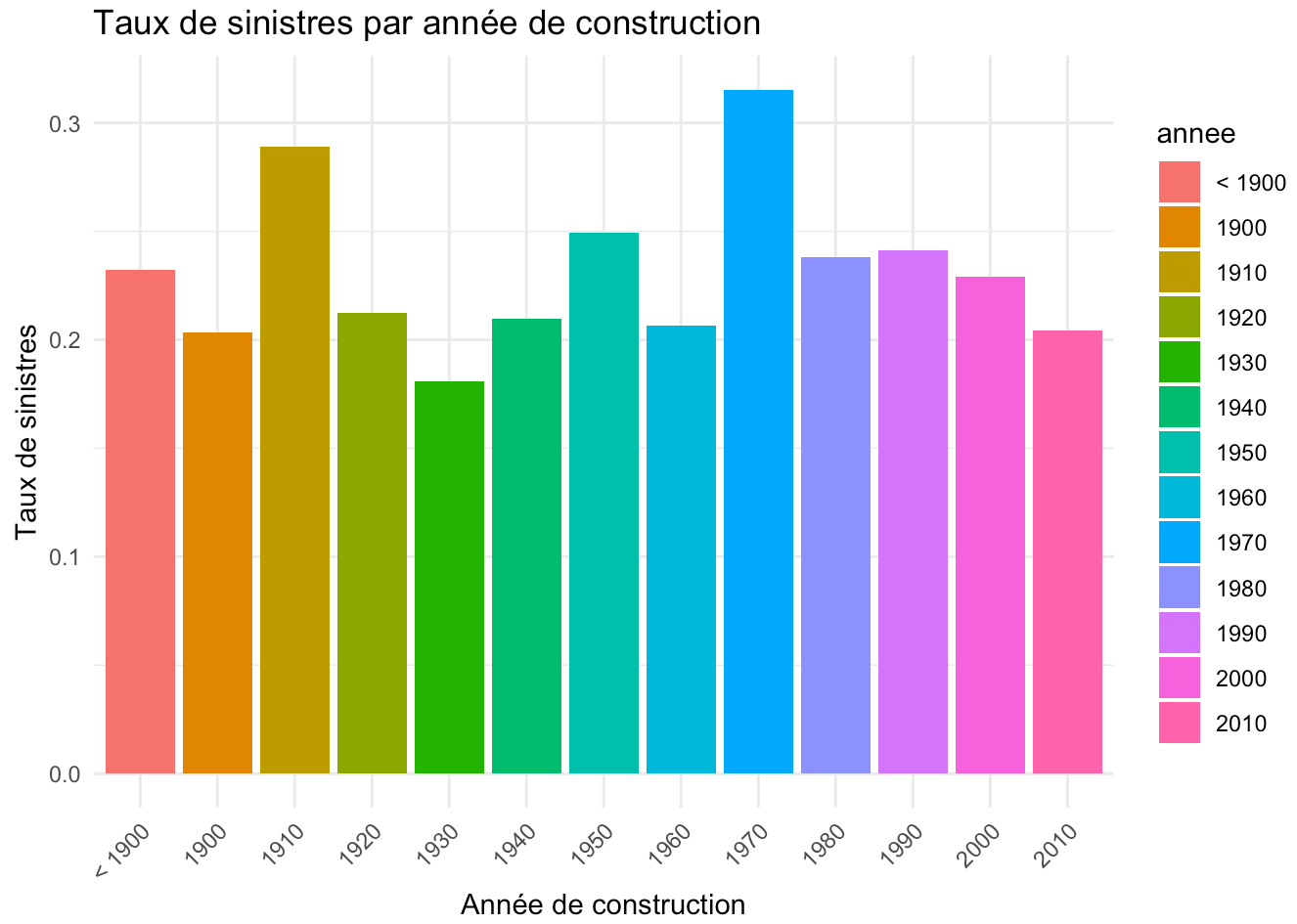
```
##  [1] "Identifiant"           "ft_2_categ"            "EXPO"
##  [4] "ft_4_categ"            "ft_5_categ"            "ft_6_categ"
##  [7] "ft_7_categ"            "ft_8_categ"            "ft_9_categ"
## [10] "ft_10_categ"           "ft_11_categ"           "ft_12_categ"
## [13] "ft_13_categ"           "ft_14_categ"           "ft_15_categ"
## [16] "ft_16_categ"           "ft_17_categ"           "ft_18_categ"
## [19] "ft_19_categ"           "superficief"           "ft_21_categ"
## [22] "ft_22_categ"           "ft_23_categ"           "ft_24_categ"
## [25] "Insee"                 "Nom_de_la_commune"     "Code_postal"
## [28] "Libellé_d_acheminement" "Numéro_Départements"   "Nom_département"
## [31] "Nom_région"
```

Vérification de l'enrichissement de X_train_enriched :

```
##  [1] "Identifiant"           "ft_2_categ"            "EXPO"
##  [4] "ft_4_categ"            "ft_5_categ"            "ft_6_categ"
##  [7] "ft_7_categ"            "ft_8_categ"            "ft_9_categ"
## [10] "ft_10_categ"           "ft_11_categ"           "ft_12_categ"
## [13] "ft_13_categ"           "ft_14_categ"           "ft_15_categ"
## [16] "ft_16_categ"           "ft_17_categ"           "ft_18_categ"
## [19] "ft_19_categ"           "superficief"           "ft_21_categ"
## [22] "ft_22_categ"           "ft_23_categ"           "ft_24_categ"
## [25] "Insee"                 "Nom_de_la_commune"     "Code_postal"
## [28] "Libellé_d_acheminement" "Numéro_Départements"   "Nom_département"
## [31] "Nom_région"
```



Nous faisons l'hypothèse que la variable ft_22_categ représente l'année de construction du bâtiment. Cette supposition est basée sur des observations préliminaires des données et pourrait avoir des implications importantes pour notre modèle, étant donné que l'âge d'un bâtiment peut influencer sa susceptibilité à subir des sinistres.



On observe que les bâtiments construits dans les années 1910 et 1970 ont le plus grand nombre de sinistres, avec des pics particulièrement élevés comparés aux autres périodes. En revanche, les bâtiments construits avant 1900 et après 2000 montrent nettement moins de sinistres.

Cela peut suggérer plusieurs interprétations:

- Les matériaux de construction ou les standards de construction pendant les années 1970 et 1980 pourraient ne pas être aussi résistants ou sécuritaires comparativement à d'autres périodes, menant à une plus grande incidence de sinistres.
- Les bâtiments construits pendant cette période pourraient maintenant être à un âge où les problèmes commencent à se manifester davantage, par rapport aux constructions plus récentes ou beaucoup plus anciennes qui ont peut-être été mieux maintenues ou rénovées.
- Des facteurs externes pourraient avoir influencé la fréquence des sinistres pour ces années de construction, comme des changements dans les réglementations, des pratiques de construction spécifiques à cette époque, ou des événements naturels qui ont eu un impact sur les bâtiments de cet âge.

Importation de données sur les catastophes naturelles (CATNAT)

L'ajout de données sur les catastrophes naturelles peut s'avérer particulièrement pertinent. Ces données peuvent enrichir notre modèle prédictif en fournissant des informations contextuelles qui ne sont pas directement liées aux caractéristiques physiques des bâtiments, mais plutôt à leur environnement et aux événements historiques.

Données sur les innondations

Nous intégrons donc des données concernant les inondations, un type de catastrophe naturelle significative en termes de fréquence et de gravité des sinistres. L'indicateur que nous utilisons est le nombre de reconnaissances de l'état de catastrophe naturelle pour les inondations publiées au Journal Officiel. Cela inclut les inondations au sens large, qui peuvent être dues à des événements comme des débordements de cours d'eau, des coulées de boue, des remontées de nappe phréatique, ou des submersions marines.

Vérification

Nous procédons à cette vérification pour nous assurer que lors de l'enrichissement des données d'entraînement avec les informations sur les inondations, chaque observation du jeu de données original a été préservée. Si le nombre de lignes reste identique entre le jeu de données original et celui enrichi après la jointure, cela indique que tous les enregistrements ont été maintenus et qu'il n'y a pas eu de perte de données. Cela validerait que l'opération de jointure a correctement associé les données supplémentaires à chaque entrée existante sans exclure ni dupliquer des lignes. Si le test renvoie "TRUE", cela confirmerait que la correspondance entre les deux jeux de données est exacte.

```
## [1] TRUE
```

```
## [1] TRUE
```

Données sur les mouvements de terrain

De la même manière que pour les données d'inondations, nous enrichissons notre jeu de données avec un indicateur spécifique aux mouvements de terrain, un autre type de risque naturel qui peut impacter significativement la probabilité de sinistres sur les bâtiments. Cet indicateur représente le nombre de reconnaissances de l'état de catastrophe naturelle pour les mouvements de terrain, publiées au Journal Officiel depuis 1982. Les mouvements de terrain comprennent divers phénomènes tels que les glissements de terrain, les effondrements, les chutes de pierres, et les tassements différentiels.

L'ajout de cet indicateur aux données de chaque commune nous permet d'intégrer une dimension temporelle longue et de capter la fréquence à laquelle une commune a été confrontée à ce type de risque naturel.

Vérification

```
## [1] TRUE
```

```
## [1] TRUE
```

Données sur les sécheresses

Pour enrichir davantage notre analyse, nous ajoutons également des données relatives aux mouvements de terrain différentiels causés par la sécheresse et la réhydratation des sols. Cet indicateur recense le nombre de reconnaissances de l'état de catastrophe naturelle spécifiquement attribuées à ce phénomène, telles qu'enregistrées au Journal Officiel depuis 1982.

Ces mouvements de terrain sont particulièrement pertinents pour les compagnies d'assurance car ils reflètent un type de sinistre qui peut causer des dommages structurels significatifs aux bâtiments. Ils sont généralement liés à des cycles climatiques qui provoquent une alternance de périodes de sécheresse et de réhydratation, entraînant des changements volumétriques dans les sols argileux qui peuvent affecter les fondations et la stabilité des constructions.

Vérification

```
## [1] TRUE
```

```
## [1] TRUE
```

Données sur les séismes

Nous poursuivons l'enrichissement de notre jeu de données avec un indicateur dédié aux séismes. Ce dernier représente le nombre de reconnaissances de l'état de catastrophe naturelle pour les séismes par commune, enregistrées au Journal Officiel depuis 1982. Les séismes sont des événements naturels d'une grande importance dans l'évaluation des risques en assurance, en raison de leur potentiel destructeur pour les structures bâties.

Vérification

```
## [1] TRUE
```

```
## [1] TRUE
```

Importation de données sur la criminalité

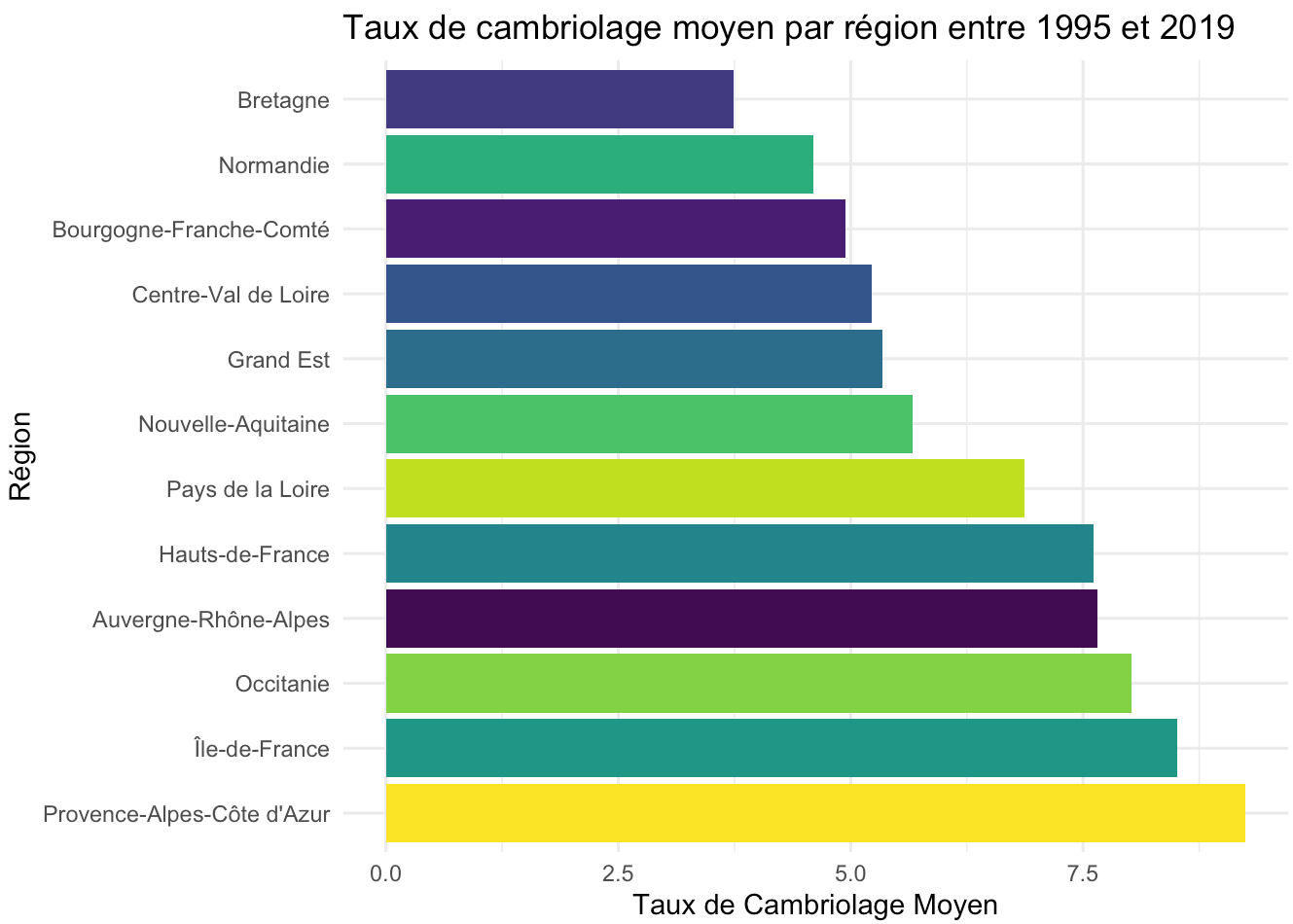
Données sur les cambriolages

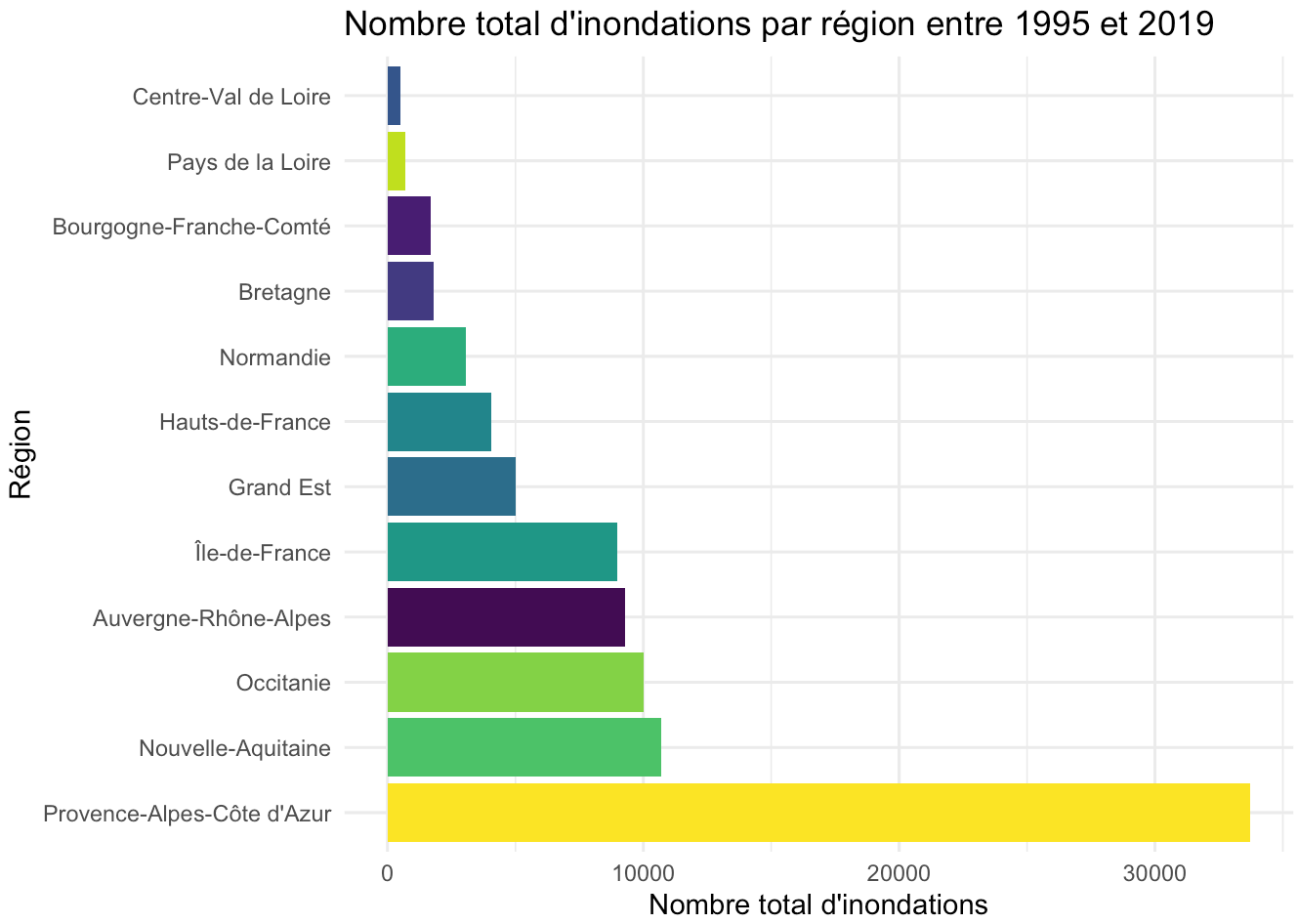
En plus des catastrophes naturelles, les données liées aux incidents de sécurité, comme les cambriolages, sont également cruciales pour l'évaluation des risques en assurance bâtiment. Nous intégrons donc un indicateur du taux moyen de cambriolage par département, basé sur les données collectées entre 2015 et 2019. Les cambriolages, en tant que sinistres potentiels, peuvent influencer considérablement le risque associé à une police d'assurance bâtiment.

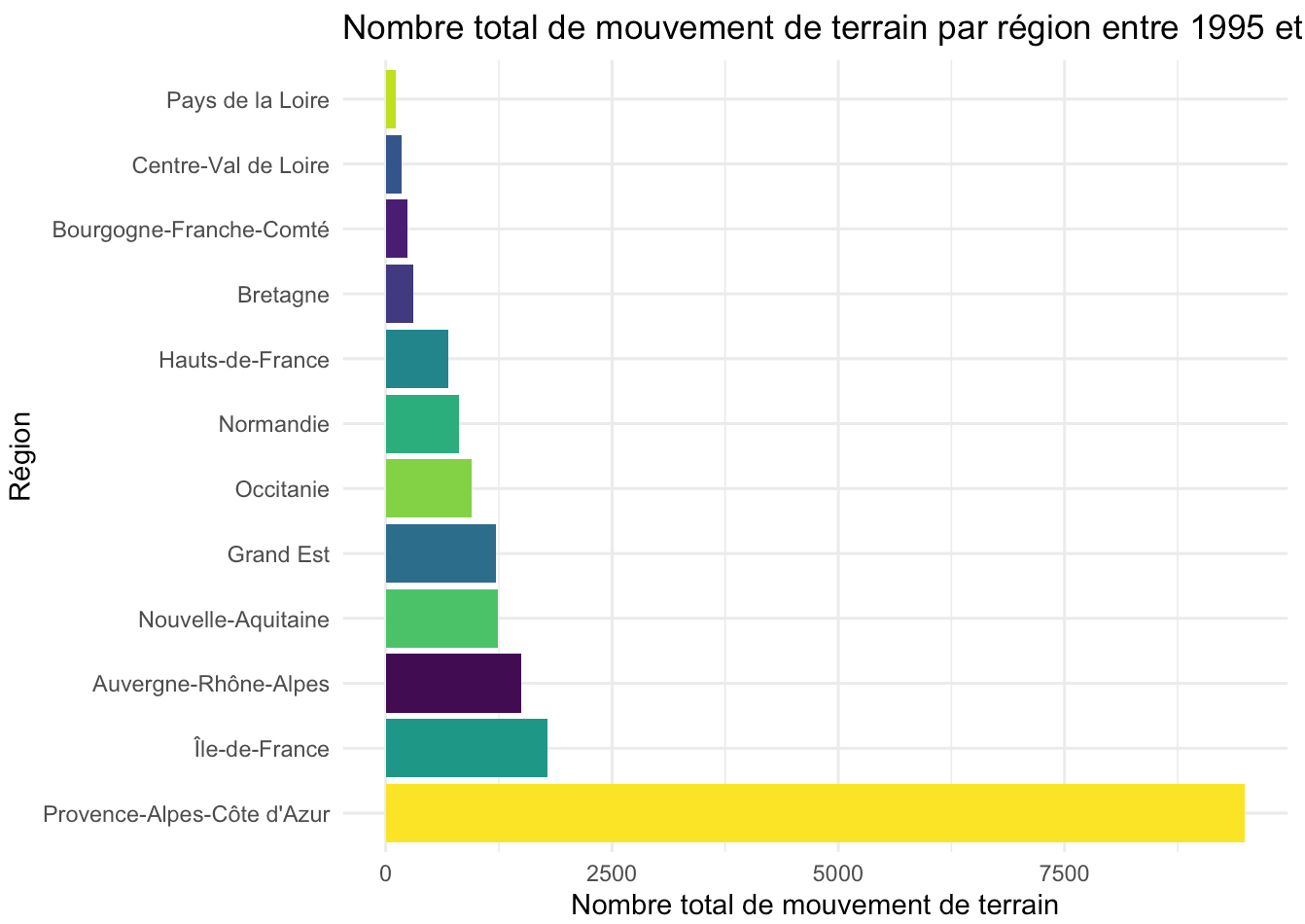
Vérification

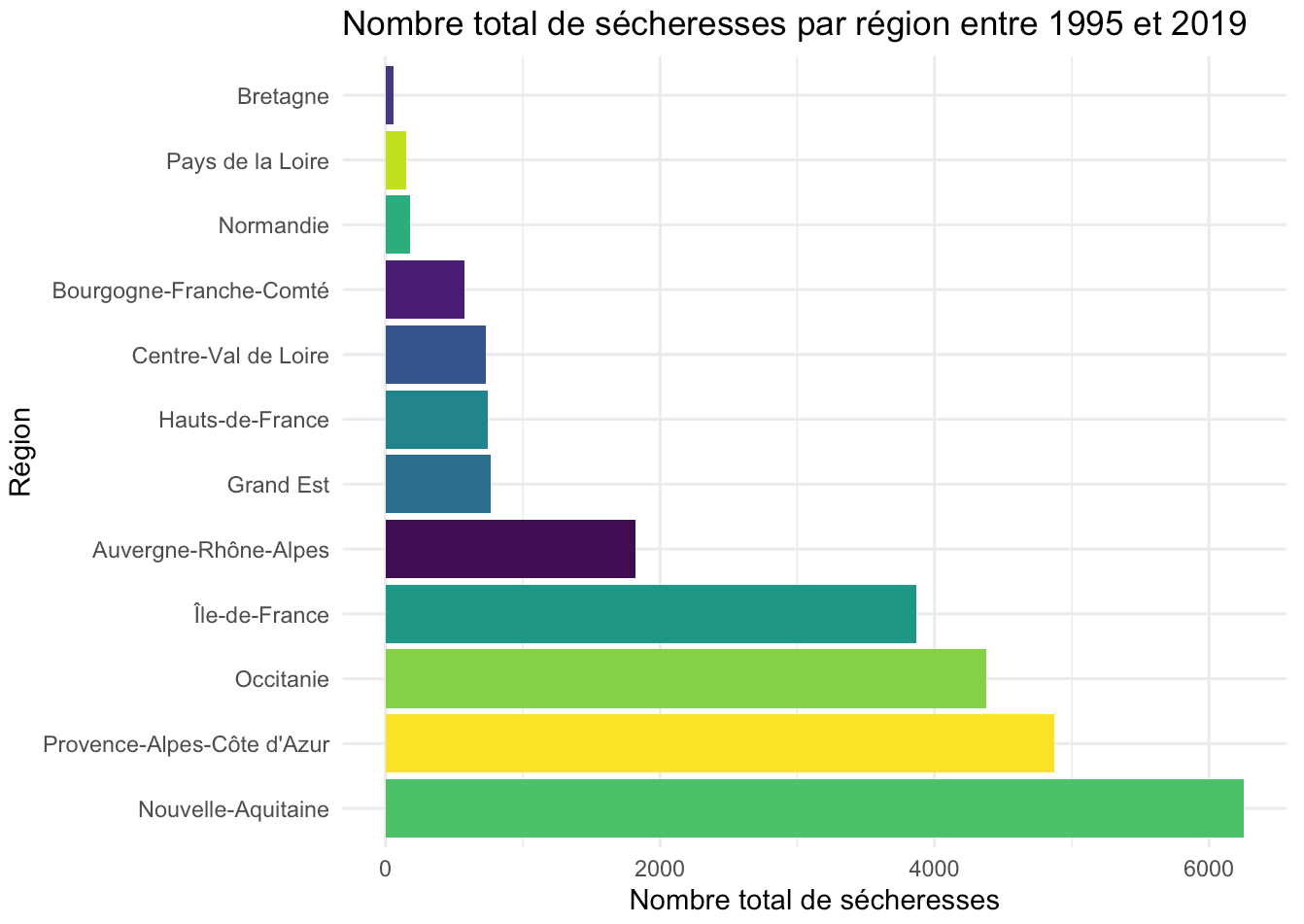
```
## [1] TRUE
```

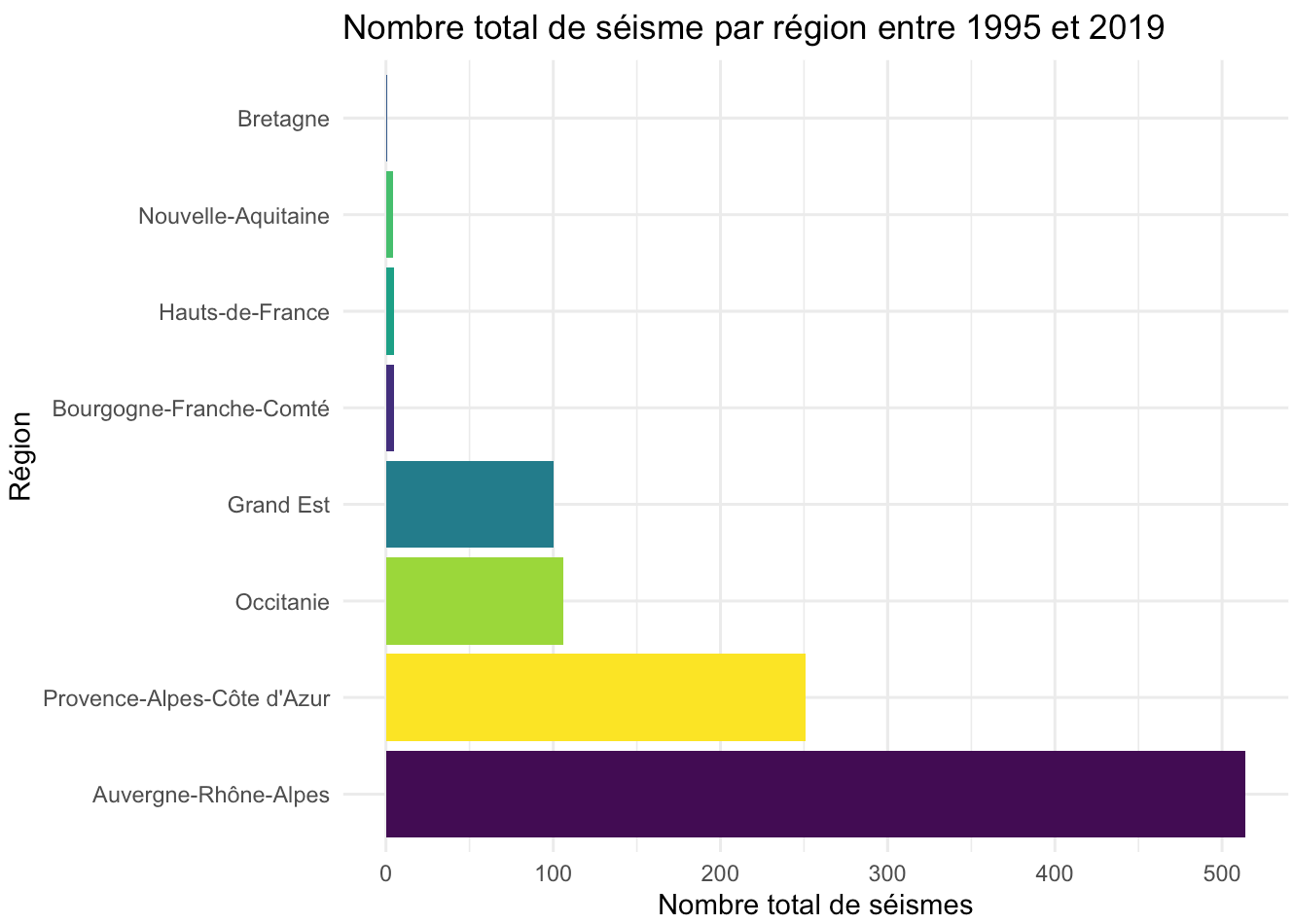
```
## [1] TRUE
```


Taux de cambriolage moyen par région entre 1995 et 2019
Bretagne
Normandie
Bourgogne-Franche-Comté
Centre-Val de Loire
Grand Est
Nouvelle-Aquitaine
Pays de la Loire
Hauts-de-France
Auvergne-Rhône-Alpes
Occitanie
Île-de-France
Provence-Alpes-Côte d'Azur
Région
0.0
2.5
5.0
7.5
Taux de Cambriolage Moyen

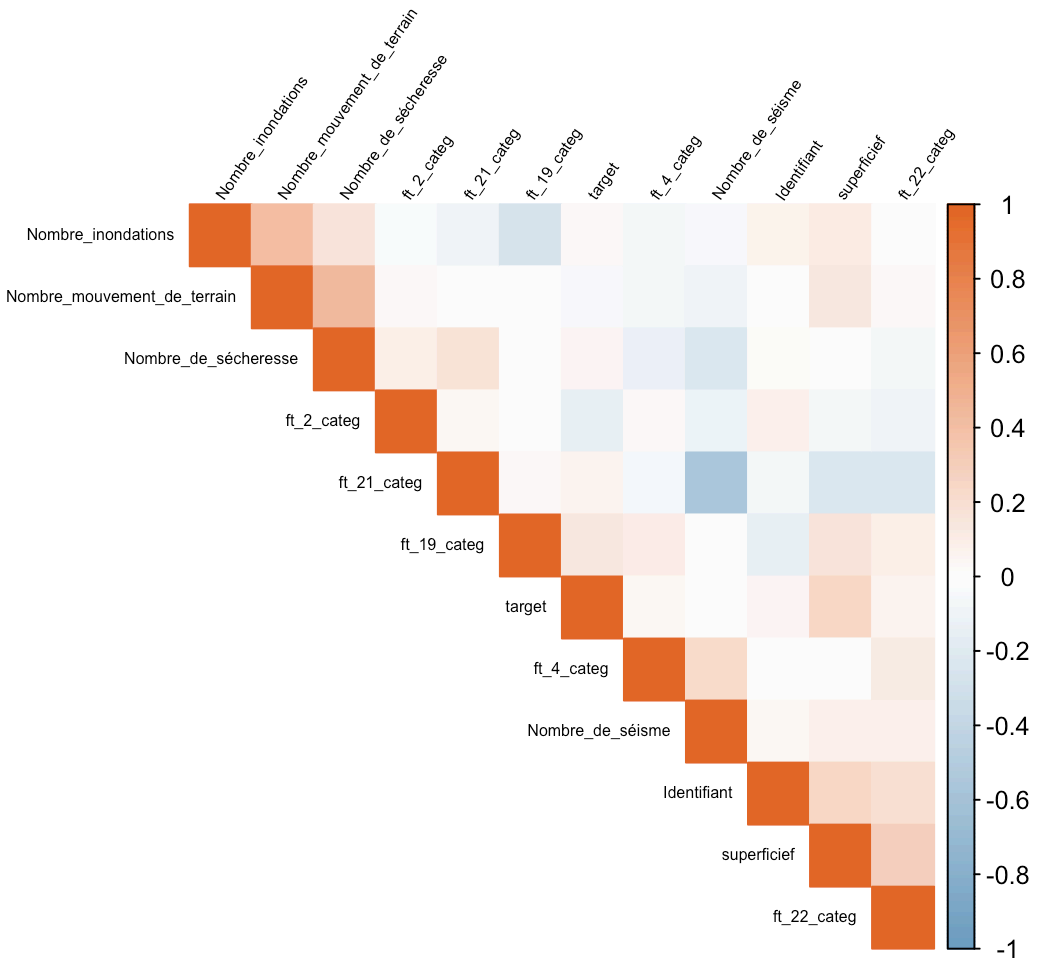

Nombre total d'inondations par région entre 1995 et 2019
Centre-Val de Loire
Pays de la Loire
Bourgogne-Franche-Comté
Bretagne
Normandie
Hauts-de-France
Grand Est
Île-de-France
Auvergne-Rhône-Alpes
Occitanie
Nouvelle-Aquitaine
Provence-Alpes-Côte d'Azur
Région
0
10000
20000
30000
Nombre total d'inondations


Nombre total de mouvement de terrain par région entre 1995 et
Pays de la Loire
Centre-Val de Loire
Bourgogne-Franche-Comté
Bretagne
Hauts-de-France
Normandie
Occitanie
Grand Est
Nouvelle-Aquitaine
Auvergne-Rhône-Alpes
Île-de-France
Provence-Alpes-Côte d'Azur
Région
0
2500
5000
7500
Nombre total de mouvement de terrain


Nombre total de sécheresses par région entre 1995 et 2019
Bretagne
Pays de la Loire
Normandie
Bourgogne-Franche-Comté
Centre-Val de Loire
Hauts-de-France
Grand Est
Auvergne-Rhône-Alpes
Île-de-France
Occitanie
Provence-Alpes-Côte d'Azur
Nouvelle-Aquitaine
Région
0
2000
4000
6000
Nombre total de sécheresses



Matrice de corrélation



Nous avons utilisé une approche spécifique pour visualiser la matrice de corrélation entre les variables numériques, y compris la variable cible. Lorsque nous avons simplement tracé la matrice de corrélation, nous avons constaté que l'affichage n'était pas optimal, ce qui rendait difficile la lecture et l'interprétation des résultats.

Pour remédier à ce problème, nous avons adopté une approche en deux étapes. Tout d'abord, nous avons utilisé la fonction corrplot pour créer un graphique de la matrice de corrélation. Ensuite, pour garantir une meilleure qualité d'affichage, nous avons sauvegardé ce graphique à la fois dans un fichier PDF en mémoire et dans un fichier PNG temporaire. Cette étape nous a permis de manipuler l'image de manière plus flexible.

Ensuite, nous avons utilisé la bibliothèque base64enc pour encoder l'image PNG en base64. Cette étape est particulièrement utile si nous voulons incorporer l'image directement notre fichier Rmd. En encodant l'image en base64, nous pouvons l'intégrer facilement dans différents contextes sans avoir à manipuler des fichiers séparés.

Notre analyse de la matrice de corrélation révèle que la majorité des variables examinées fournissent des contributions distinctes au modèle. Ce constat est prometteur car il indique un minimum de chevauchement informationnel entre les variables, réduisant ainsi le risque de multicollinéarité lors de la modélisation prédictive. Cela signifie que chaque variable a le potentiel d'améliorer la robustesse du modèle sans empiéter sur l'information déjà apportée par une autre.

En particulier, la corrélation observée entre la variable superficief et la variable cible target est la plus élevée. Il est logique de supposer qu'un bâtiment de plus grande superficie ait un risque plus élevé de sinistre, puisqu'il y a davantage de surface susceptible d'être impactée.

Modélisation

1. Arbre de classification

Le premier modèle utilise un arbre de classification pour prédire la probabilité qu'un bâtiment dépose une réclamation d'assurance pendant une période donnée, en se basant sur ses caractéristiques.

Étape 1 : Préparation des données

Nous avons réparti notre jeu de données initial en deux sous-ensembles : 80% pour l'entraînement et 20% pour le test. Cette séparation a été réalisée en stratifiant selon la variable cible pour garantir une distribution équilibrée des classes dans chaque sous-ensemble.

Étape 2 : Prétraitement des données

Nous avons identifié les variables catégorielles comme celles ayant moins de 20 valeurs uniques et les avons converties en facteurs dans l'ensemble d'entraînement, ce qui est nécessaire pour la construction de l'arbre de décision.

Nous avons renommé plusieurs colonnes complexes en noms plus courts et plus compréhensibles, qui reflètent les différents types de catastrophes naturelles. Cette étape est cruciale pour faciliter les manipulations et les analyses ultérieures.

Les valeurs de la colonne EXPO, exprimées initialement avec des virgules comme séparateurs décimaux, ont été converties en format numérique en remplaçant les virgules par des points, puis en les convertissant en type numérique. Nous avons également vérifié et imputé les valeurs manquantes par la médiane pour cette colonne et d'autres colonnes numériques.

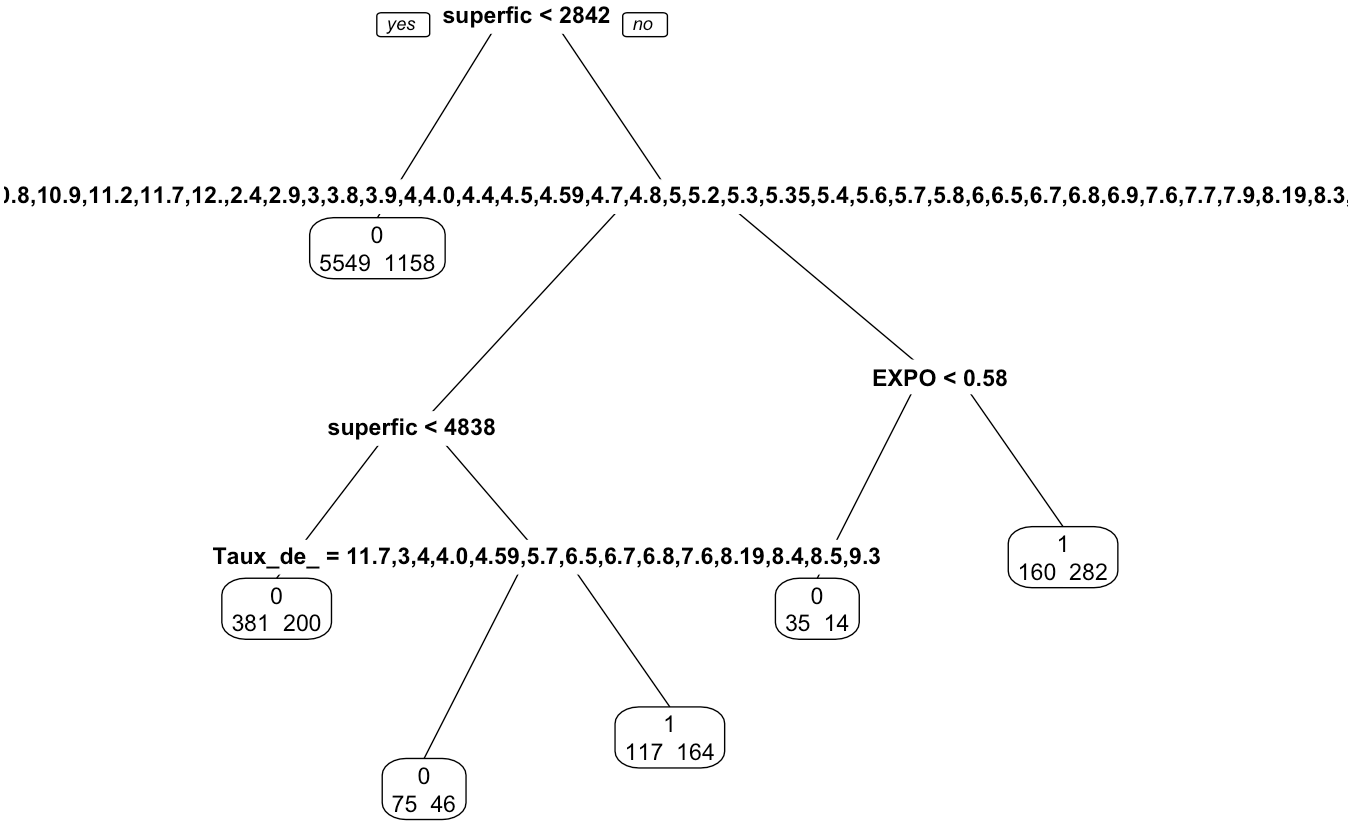
Dans le traitement des ensembles de données X_train_enriched et X_test_enriched, nous choisissons de retirer certaines colonnes qui ne sont pas utiles pour notre modèle. Les variables supprimées sont principalement des identifiants géographiques et des labels qui pourraient introduire du bruit ou un biais dans notre modèle, ou qui sont tout simplement redondantes.

Enfin, nous ajustons la colonne Taux_de_Cambriolage pour garantir que les données sont numériques et uniformément formatées (remplacement des virgules par des points et suppression des caractères non numériques). Nous calculons la médiane pour les valeurs de cambriolage et utilisons cette médiane pour imputer les valeurs manquantes dans cette colonne.

Étape 3 : Construction et évaluation de l'arbre de décision

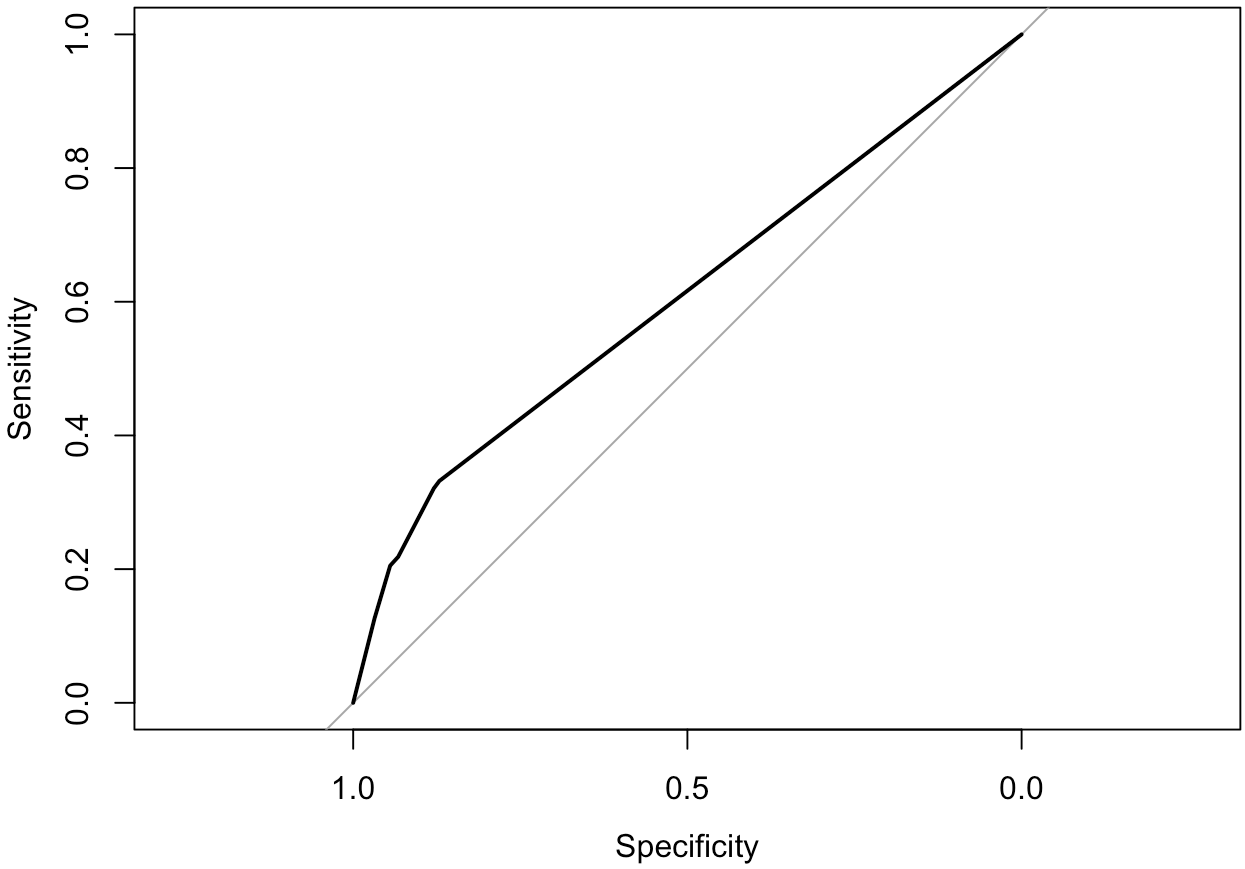
Nous avons ensuite construit un arbre de décision en utilisant la fonction rpart. Pour comprendre l'importance des différentes variables et la structure de l'arbre, nous avons visualisé celui-ci avec la fonction prp, en montrant les informations supplémentaires telles que le gain d'information à chaque nœud.

Nous avons évalué les performances de notre modèle sur les ensembles d'apprentissage et de test à l'aide d'une matrice de confusion, nous permettant de calculer des mesures telles que la précision, la sensibilité, la spécificité, et la valeur prédictive.



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6040 1418
##          1  277  446
##
##                Accuracy : 0.7928
##                  95% CI : (0.7839, 0.8016)
##     No Information Rate : 0.7722
##     P-Value [Acc > NIR] : 3.588e-06
##
##                   Kappa : 0.2492
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9562
##             Specificity : 0.2393
##          Pos Pred Value : 0.8099
##          Neg Pred Value : 0.6169
##              Prevalence : 0.7722
##          Detection Rate : 0.7383
##    Detection Prevalence : 0.9116
##       Balanced Accuracy : 0.5977
##
##        'Positive' Class : 0
##
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1502  364
##          1   88   94
##
##                Accuracy : 0.7793
##                  95% CI : (0.7607, 0.7971)
##     No Information Rate : 0.7764
##     P-Value [Acc > NIR] : 0.387
##
##                   Kappa : 0.1908
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9447
##             Specificity : 0.2052
##          Pos Pred Value : 0.8049
##          Neg Pred Value : 0.5165
##              Prevalence : 0.7764
##          Detection Rate : 0.7334
##    Detection Prevalence : 0.9111
##       Balanced Accuracy : 0.5749
##
##        'Positive' Class : 0
##
```



Le modèle présente une sensibilité élevée mais une spécificité très faible, ce qui signifie qu'il est bon pour détecter les cas positifs mais tend à mal classer de nombreux cas négatifs comme positifs. La valeur Kappa faible dans les deux cas indique également que la précision du modèle n'est pas beaucoup meilleure que le hasard, en particulier dans des situations où le taux de prévalence de l'une des classes est élevé.

Étape 4 : Analyse de la performance du modèle

Pour quantifier la capacité de notre modèle à distinguer entre les différentes classes, nous avons calculé l'AUC et, en conséquence, le score Gini normalisé. Ces mesures nous donnent une indication de la qualité de notre modèle dans le contexte de la prédiction des réclamations d'assurance pour les bâtiments.

```
## Area under the curve: 0.6056
```

```
## [1] "Le score Gini pour le modèle de l'arbre de décision est : 0.211172447886627"
```

Étape 5 : Prédiction des probabilités

Après avoir entraîné notre arbre de décision et évalué ses performances avec l'AUC et le score Gini, nous avons appliqué le modèle à notre ensemble de données de test enrichi pour prédire les probabilités de réclamations d'assurance. En utilisant la fonction predict et en spécifiant le type de sortie comme "prob", notre modèle a généré les probabilités correspondantes pour chaque bâtiment dans l'ensemble de test, nous donnant ainsi des informations précieuses sur le risque associé à chaque propriété.

Afin de participer au Data Challenge de l'ENS, nous avons préparé un fichier de soumission comprenant les probabilités estimées de sinistre pour chaque bâtiment. Ce fichier aligne les identifiants des bâtiments avec les probabilités correspondantes, suivant le format requis par la plateforme du challenge.

La soumission de ce fichier sur le site du Data Challenge ENS a permis de convertir nos prédictions en un score de 0,1756.

Conclusion

L'analyse des performances de notre arbre de décision a fourni des insights pertinents sur ses capacités prédictives. Avec un AUC de 0.6056, le modèle démontre une certaine aptitude à distinguer les bâtiments susceptibles de subir un sinistre de ceux qui ne le sont pas, bien que l'espace pour améliorer sa discrimination soit notable. Le score Gini normalisé de 0.2111 renforce cette observation, indiquant que, bien que le modèle soit meilleur que le hasard, il y a une marge significative d'amélioration.

Fort de ces conclusions, nous avons entrepris de développer un deuxième modèle prédictif en utilisant l'algorithme avancé XGBoost. XGBoost, ou eXtreme Gradient Boosting, est une méthode d'apprentissage automatique qui s'est imposée dans les domaines du machine learning pour sa performance dans les tâches de classification et de régression. Ce modèle promet d'exploiter la structure des données d'une manière plus raffinée et potentiellement plus efficace grâce à ses capacités d'apprentissage en profondeur.

Transitionnant de l'approche de modélisation basée sur un arbre unique, XGBoost nous permet de combiner les prédictions de nombreux arbres de décision, potentiellement améliorant la robustesse et l'exactitude des prédictions. Notre objectif avec ce deuxième modèle est de capturer des interactions plus complexes entre les variables et de fournir une prédiction plus nuancée des probabilités de sinistre pour chaque bâtiment.

Dans les étapes suivantes, nous détaillons notre approche pour préparer les données pour XGBoost, optimiser ses paramètres et évaluer ses performances, visant à surpasser le modèle initial d'arbre de décision.

2. XGBoost

Nous a développé un deuxième modèle prédictif utilisant XGBoost pour estimer la probabilité qu'un bâtiment fasse l'objet d'une réclamation d'assurance durant une période donnée. Voici comment nous avons procédé et les résultats que nous avons obtenus.

Étape 1: Préparation des données

Nous avons commencé par partitionner les données en un ensemble d'entraînement (80%) et un ensemble de test (20%), en utilisant une fonction de partitionnement pour garantir la reproductibilité des résultats grâce à l'initialisation du générateur de nombres aléatoires (set.seed(1)).

Nous avons décidé d'exclure des colonnes spécifiques comme Libellé_d_acheminement, Insee, Nom_de_la_commune, et autres identificateurs qui ne sont pas utiles pour la modélisation. Ces informations sont principalement des identifiants et des descriptions textuelles qui n'apportent pas de valeur prédictive significative à notre modèle.

Les données EXPO contenaient des valeurs numériques formatées avec des virgules pour les décimales, que nous avons converties en points pour uniformiser le format numérique, suivi de la conversion en type numérique réel (as.numeric).

Pour les colonnes numériques, nous avons remplacé les valeurs manquantes par la médiane, qui est moins sensible aux valeurs aberrantes que la moyenne, assurant ainsi une meilleure robustesse des données.

Nous ajustons la colonne Taux_de_Cambriolage pour garantir que les données sont numériques et uniformément formatées (remplacement des virgules par des points et suppression des caractères non numériques). Nous calculons la médiane pour les valeurs de cambriolage et utilisons cette médiane pour imputer les valeurs manquantes dans cette colonne.

Pour les variables catégorielles comme ft_8_categ, nous les avons converties en facteurs puis en numériques, attribuant des entiers uniques à chaque catégorie.

Nous avons donc converti toutes les colonnes en valeurs numériques, ce qui est une exigence pour utiliser efficacement l'algorithme de machine learning XGBoost, qui manipule des données numériques.

En préparant les données de cette manière, nous assurons que notre modèle XGBoost peut être entraîné efficacement, en utilisant des caractéristiques pertinentes et bien formatées, optimisant ainsi la précision de la prédiction sur les données de test.

Étape 2: Configuration du modèle XGBoost

Pour optimiser les hyperparamètres de XGBoost, nous avons utilisé une grille de recherche (tuneGrid) avec des paramètres prédéfinis. Ces paramètres incluaient le nombre de cycles de boosting (nrounds), la profondeur maximale des arbres (max_depth), le poids minimal des enfants (min_child_weight), la fraction d'échantillons utilisés pour entraîner chaque arbre (subsample), la fraction des colonnes à utiliser par arbre (colsample_bytree), le taux d'apprentissage (eta), et le paramètre de régularisation (gamma).

Nous avons ensuite utilisé une validation croisée à 10 plis pour évaluer les modèles, ce qui a permis de minimiser le sur-ajustement et d'améliorer la robustesse du modèle final.

Étape 3: Résultats et validation du modèle

Le meilleur ensemble de paramètres a été automatiquement sélectionné par la grille de recherche. Avec ces paramètres optimisés, nous avons entraîné le modèle final, utilisé pour prédire les probabilités de sinistre sur l'ensemble de test. Nous avons également produit une matrice d'importance pour identifier les variables les plus influentes dans les prédictions du modèle, et nous avons visualisé les premiers arbres du modèle pour mieux comprendre les décisions prises par le modèle.

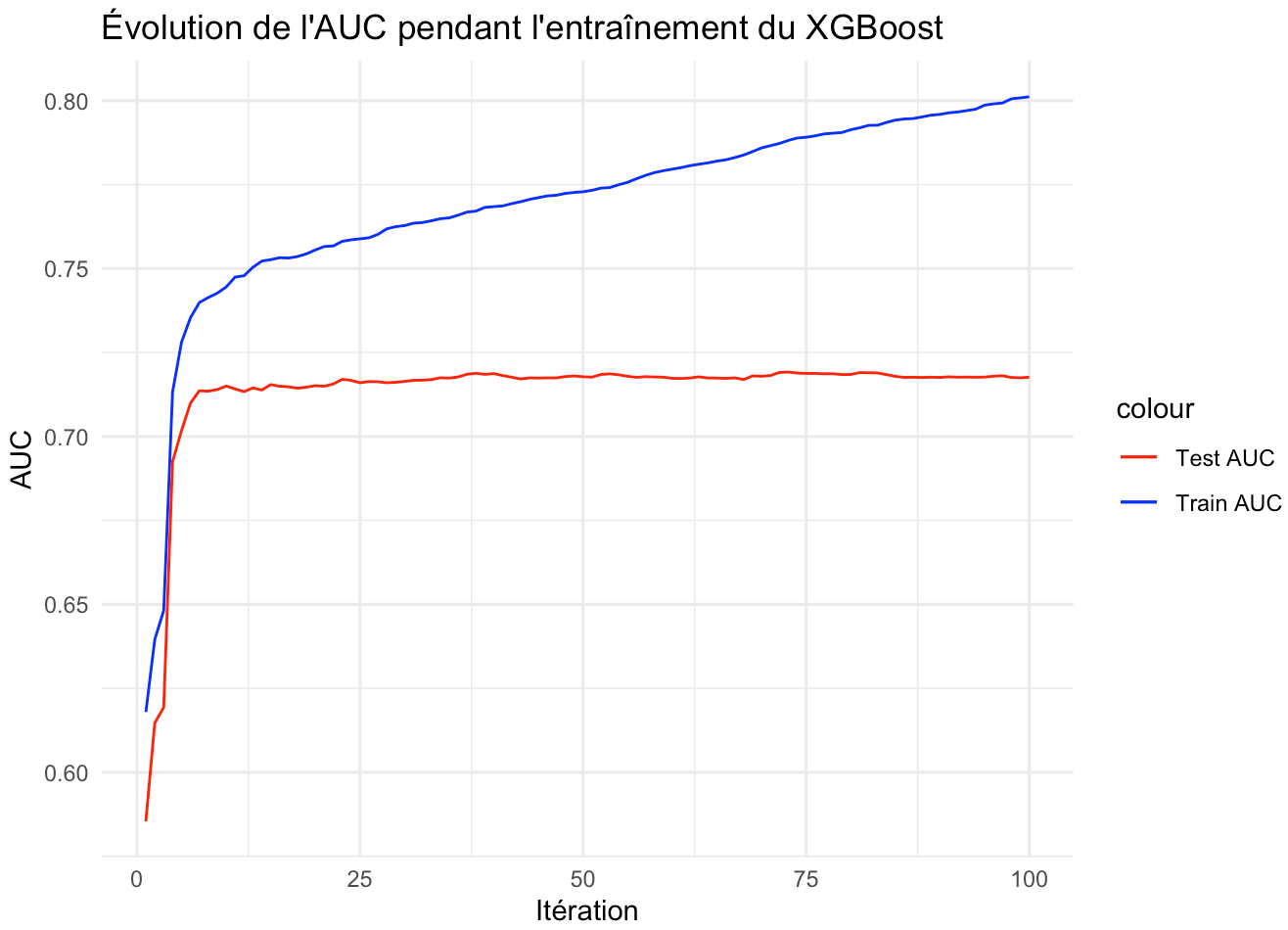
Meilleurs hyperparamètres

```
##   nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 1      50         4 0.1     0              0.7                1       0.8
```

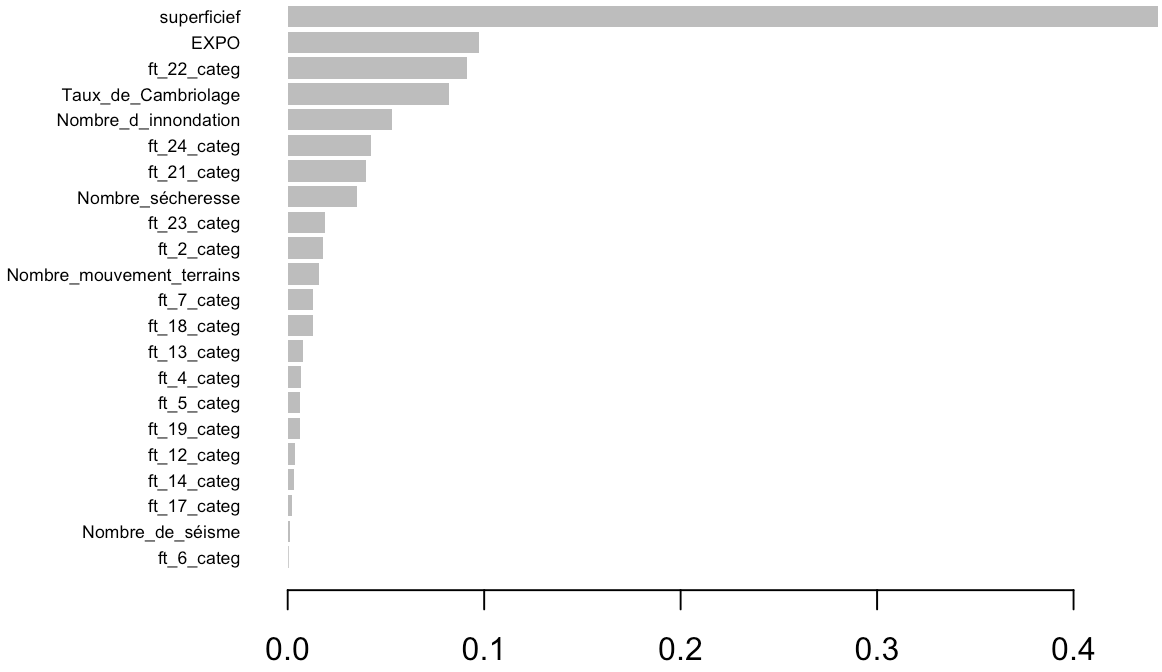
Entraînement du modèle

```
## [1]  train-auc:0.617951  test-auc:0.585399
## [2]  train-auc:0.639577  test-auc:0.614685
## [3]  train-auc:0.648166  test-auc:0.619342
## [4]  train-auc:0.713357  test-auc:0.692575
## [5]  train-auc:0.728109  test-auc:0.701737
## [6]  train-auc:0.735368  test-auc:0.709934
## [7]  train-auc:0.739899  test-auc:0.713630
## [8]  train-auc:0.741376  test-auc:0.713486
## [9]  train-auc:0.742668  test-auc:0.713964
## [10] train-auc:0.744505  test-auc:0.715011
## [11] train-auc:0.747454  test-auc:0.714159
## [12] train-auc:0.747856  test-auc:0.713351
## [13] train-auc:0.750375  test-auc:0.714449
## [14] train-auc:0.752229  test-auc:0.713834
## [15] train-auc:0.752641  test-auc:0.715452
## [16] train-auc:0.753237  test-auc:0.714941
## [17] train-auc:0.753126  test-auc:0.714793
## [18] train-auc:0.753580  test-auc:0.714366
## [19] train-auc:0.754381  test-auc:0.714680
## [20] train-auc:0.755541  test-auc:0.715124
## [21] train-auc:0.756571  test-auc:0.714962
## [22] train-auc:0.756716  test-auc:0.715620
## [23] train-auc:0.758096  test-auc:0.717009
## [24] train-auc:0.758566  test-auc:0.716705
## [25] train-auc:0.758844  test-auc:0.715990
## [26] train-auc:0.759152  test-auc:0.716350
## [27] train-auc:0.760170  test-auc:0.716285
## [28] train-auc:0.761817  test-auc:0.715996
## [29] train-auc:0.762465  test-auc:0.716129
## [30] train-auc:0.762794  test-auc:0.716401
## [31] train-auc:0.763513  test-auc:0.716718
## [32] train-auc:0.763694  test-auc:0.716761
## [33] train-auc:0.764226  test-auc:0.716924
## [34] train-auc:0.764837  test-auc:0.717479
## [35] train-auc:0.765081  test-auc:0.717377
## [36] train-auc:0.765904  test-auc:0.717718
## [37] train-auc:0.766810  test-auc:0.718535
## [38] train-auc:0.767110  test-auc:0.718803
## [39] train-auc:0.768212  test-auc:0.718485
## [40] train-auc:0.768432  test-auc:0.718719
## [41] train-auc:0.768651  test-auc:0.718124
## [42] train-auc:0.769324  test-auc:0.717652
## [43] train-auc:0.769895  test-auc:0.717120
## [44] train-auc:0.770589  test-auc:0.717452
## [45] train-auc:0.771128  test-auc:0.717384
## [46] train-auc:0.771630  test-auc:0.717457
## [47] train-auc:0.771822  test-auc:0.717444
## [48] train-auc:0.772361  test-auc:0.717839
## [49] train-auc:0.772658  test-auc:0.718016
## [50] train-auc:0.772866  test-auc:0.717774
## [51] train-auc:0.773290  test-auc:0.717656
## [52] train-auc:0.773956  test-auc:0.718448
## [53] train-auc:0.774114  test-auc:0.718650
## [54] train-auc:0.774962  test-auc:0.718356
## [55] train-auc:0.775689  test-auc:0.717914
## [56] train-auc:0.776759  test-auc:0.717604
```

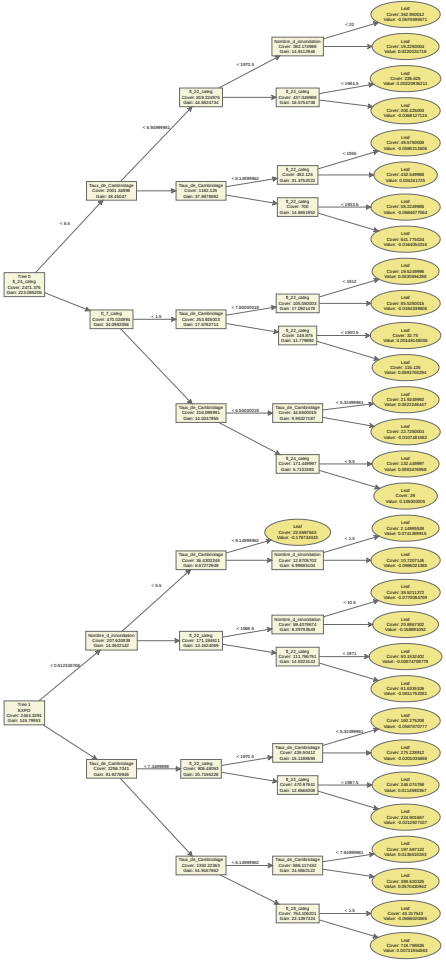
```
## [57] train-auc:0.777736  test-auc:0.717807
## [58] train-auc:0.778569  test-auc:0.717719
## [59] train-auc:0.779129  test-auc:0.717654
## [60] train-auc:0.779602  test-auc:0.717316
## [61] train-auc:0.780058  test-auc:0.717257
## [62] train-auc:0.780646  test-auc:0.717424
## [63] train-auc:0.781064  test-auc:0.717755
## [64] train-auc:0.781466  test-auc:0.717420
## [65] train-auc:0.781975  test-auc:0.717388
## [66] train-auc:0.782388  test-auc:0.717290
## [67] train-auc:0.783046  test-auc:0.717453
## [68] train-auc:0.783803  test-auc:0.716950
## [69] train-auc:0.784831  test-auc:0.718033
## [70] train-auc:0.785918  test-auc:0.717925
## [71] train-auc:0.786587  test-auc:0.718141
## [72] train-auc:0.787248  test-auc:0.719061
## [73] train-auc:0.788150  test-auc:0.719182
## [74] train-auc:0.788868  test-auc:0.718886
## [75] train-auc:0.789087  test-auc:0.718774
## [76] train-auc:0.789508  test-auc:0.718774
## [77] train-auc:0.790079  test-auc:0.718665
## [78] train-auc:0.790296  test-auc:0.718679
## [79] train-auc:0.790496  test-auc:0.718425
## [80] train-auc:0.791342  test-auc:0.718468
## [81] train-auc:0.791902  test-auc:0.719021
## [82] train-auc:0.792636  test-auc:0.718969
## [83] train-auc:0.792671  test-auc:0.718913
## [84] train-auc:0.793503  test-auc:0.718416
## [85] train-auc:0.794202  test-auc:0.717898
## [86] train-auc:0.794535  test-auc:0.717594
## [87] train-auc:0.794664  test-auc:0.717637
## [88] train-auc:0.795148  test-auc:0.717559
## [89] train-auc:0.795666  test-auc:0.717647
## [90] train-auc:0.795904  test-auc:0.717572
## [91] train-auc:0.796375  test-auc:0.717770
## [92] train-auc:0.796622  test-auc:0.717646
## [93] train-auc:0.797029  test-auc:0.717678
## [94] train-auc:0.797449  test-auc:0.717634
## [95] train-auc:0.798638  test-auc:0.717678
## [96] train-auc:0.799039  test-auc:0.717943
## [97] train-auc:0.799282  test-auc:0.718084
## [98] train-auc:0.800534  test-auc:0.717560
## [99] train-auc:0.800821  test-auc:0.717478
## [100]    train-auc:0.801150  test-auc:0.717626
```



Matrice d’importance



Graphique de l’arbre



Étape 4: Évaluation des performances

Pour évaluer les performances de notre modèle, nous avons calculé l'AUC (Area Under the Curve) et le coefficient de Gini normalisé.

```
## [1] "AUC: 0.717625904111164"
```

```
## [1] "Le score Gini pour le modèle XGBoost : 0.435251808222329"
```

Étape 5: Prédiction sur l'ensemble de test

Nous avons complété notre analyse en effectuant des prédictions sur l'ensemble de test. Cette étape finale utilise le modèle final pour prédire les probabilités que chaque bâtiment dans l'ensemble de test dépose une réclamation d'assurance.

Nous avons préparé un fichier de soumission comprenant les probabilités estimées de sinistre pour chaque bâtiment. La soumission de ce fichier sur le site du Data Challenge ENS a permis de convertir nos prédictions en un score de 0,3960.

Conclusion

L'approche XGBoost a démontré son efficacité par rapport au modèle précédent de l'arbre de décision, avec un score AUC nettement amélioré de 0.7176, ce qui reflète une meilleure capacité à différencier les bâtiments susceptibles de soumettre une réclamation d'assurance. Le score Gini normalisé a doublé à 0.4352, indiquant que le modèle XGBoost est nettement plus discriminant que le modèle basé sur un seul arbre de classification.

Les améliorations apportées par XGBoost résident dans sa capacité à combiner les prédictions de multiples arbres de décision, permettant ainsi de capturer des nuances plus complexes au sein des données. L'interprétation des variables importantes et la visualisation des arbres contribuent à une compréhension transparente des facteurs influençant les prédictions. En outre, le modèle a été validé à travers un processus rigoureux de validation croisée, renforçant notre confiance dans sa capacité à généraliser au-delà de l'ensemble d'entraînement.

Dans le cadre de notre étude avec le modèle XGBoost un aspect crucial a été l'optimisation des hyperparamètres. Ici, pour des raisons computationnelles, notre grille de recherche (tuneGrid) a été intentionnellement simplifiée, ne contenant qu'une seule valeur pour chaque paramètre. Cette approche visait à réduire la charge et le temps de calcul nécessaires pour trouver une configuration performante. Avec cette configuration de base, nous avons atteint un score de 0,4328 sur le site de l'ENS, un résultat prometteur compte tenu de la simplicité du modèle.

Cependant, conscients du potentiel de l'optimisation pour améliorer davantage les performances, nous avons élargi notre grille de recherche. Le grid search était défini comme suit :

tuneGrid <- expand.grid(nrounds = c(50,100,150), max_depth = c(4,8,16,32), min_child_weight = c(1), subsample = c(0.8), colsample_bytree = c(0.7), eta = c(0.1,0.01,0.001), gamma = c(0))

Cette approche plus exhaustive permet de tester une variété plus large de combinaisons, donnant au modèle la possibilité d'explorer divers scénarios de complexité et d'adaptabilité. En utilisant ce grid search élargi, nous avons pu améliorer le score de précision sur le site de l'ENS à 0,4400.

Alors que nous progressons vers l'implémentation de LightGBM (Light Gradient Boosting Machine), nous nous attendons à explorer d'autres façons d'optimiser la performance. LightGBM est connu pour sa rapidité et son efficacité sur les grands ensembles de données et sa capacité à gérer les variables catégorielles de manière native. Ces caractéristiques pourraient s'avérer avantageuses dans notre contexte, étant donné la nature de nos données et l'objectif de prédiction.

Dans les prochaines étapes, nous adapterons nos données aux spécifications de LightGBM et affinerons la configuration de notre modèle pour exploiter pleinement ses capacités. Nous évaluerons ensuite la performance du modèle LightGBM avec l'espoir d'améliorer encore les résultats obtenus avec XGBoost.

3. LGMBoost

Étape 1 : Préparation des données

Nous avons préparé nos jeux de données pour l'entraînement et le test en les convertissant en matrices et en traitant les variables catégorielles, puisque LightGBM gère nativement les caractéristiques catégorielles.

Pour les colonnes numériques, nous avons imputé les valeurs manquantes par la médiane garantissant ainsi que nos modèles ne soient pas biaisés ou affectés par des erreurs dues à des valeurs manquantes.

Étape 2 : Configuration du modèle LightGBM

Nous avons défini les paramètres de LightGBM en choisissant un objectif binaire, puisque nous effectuons une classification. Nous avons opté pour une optimisation basée sur l'AUC avec des paramètres tels que le nombre de feuilles, le taux d'apprentissage et le nombre d'estimateurs, entre autres, pour contrôler la complexité du modèle et éviter le sur-ajustement.

Étape 3 : Entraînement du modèle et évaluation

En utilisant les données d'entraînement, nous avons entraîné le modèle avec 800 cycles de boosting, tout en évaluant les performances à la fois sur l'ensemble d'entraînement et de test pour surveiller l'évolution de l'AUC. Cette approche nous a permis de détecter et d'éviter le sur-ajustement.

```
## [LightGBM] [Info] Number of positive: 1838, number of negative: 6346
## [LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.000617
seconds.
## You can set `force_row_wise=true` to remove the overhead.
## And if memory is not enough, you can set `force_col_wise=true`.
## [LightGBM] [Info] Total Bins 959
## [LightGBM] [Info] Number of data points in the train set: 8184, number of used features: 17
## [LightGBM] [Info] [binary:BoostFromScore]: pavg=0.224585 -> initscore=-1.239147
## [LightGBM] [Info] Start training from score -1.239147
## [1]:  train's auc:0.713943  test's auc:0.693123
## [2]:  train's auc:0.716112  test's auc:0.694366
## [3]:  train's auc:0.724614  test's auc:0.703037
## [4]:  train's auc:0.72699  test's auc:0.706441
## [5]:  train's auc:0.726658  test's auc:0.706631
## [6]:  train's auc:0.731115  test's auc:0.710364
## [7]:  train's auc:0.734219  test's auc:0.71153
## [8]:  train's auc:0.738358  test's auc:0.716388
## [9]:  train's auc:0.743107  test's auc:0.718201
## [10]:  train's auc:0.742663  test's auc:0.717801
## [11]:  train's auc:0.744657  test's auc:0.719031
## [12]:  train's auc:0.745073  test's auc:0.720313
## [13]:  train's auc:0.747891  test's auc:0.721379
## [14]:  train's auc:0.748562  test's auc:0.720964
## [15]:  train's auc:0.748568  test's auc:0.720651
## [16]:  train's auc:0.750172  test's auc:0.721413
## [17]:  train's auc:0.750775  test's auc:0.722176
## [18]:  train's auc:0.751165  test's auc:0.722799
## [19]:  train's auc:0.752332  test's auc:0.721768
## [20]:  train's auc:0.752422  test's auc:0.721988
## [21]:  train's auc:0.752682  test's auc:0.721783
## [22]:  train's auc:0.752951  test's auc:0.721419
## [23]:  train's auc:0.753138  test's auc:0.721115
## [24]:  train's auc:0.753307  test's auc:0.720888
## [25]:  train's auc:0.753338  test's auc:0.720651
## [26]:  train's auc:0.754757  test's auc:0.721185
## [27]:  train's auc:0.755015  test's auc:0.721541
## [28]:  train's auc:0.755909  test's auc:0.721512
## [29]:  train's auc:0.756596  test's auc:0.722375
## [30]:  train's auc:0.756963  test's auc:0.722619
## [31]:  train's auc:0.757446  test's auc:0.722039
## [32]:  train's auc:0.757516  test's auc:0.721941
## [33]:  train's auc:0.757713  test's auc:0.721857
## [34]:  train's auc:0.758369  test's auc:0.721452
## [35]:  train's auc:0.758383  test's auc:0.721396
## [36]:  train's auc:0.758691  test's auc:0.722092
## [37]:  train's auc:0.758805  test's auc:0.722565
## [38]:  train's auc:0.758913  test's auc:0.722894
## [39]:  train's auc:0.759094  test's auc:0.723021
## [40]:  train's auc:0.7592  test's auc:0.723347
## [41]:  train's auc:0.759774  test's auc:0.723232
## [42]:  train's auc:0.759897  test's auc:0.723644
## [43]:  train's auc:0.760079  test's auc:0.723694
## [44]:  train's auc:0.760261  test's auc:0.723868
## [45]:  train's auc:0.760348  test's auc:0.724142
## [46]:  train's auc:0.761033  test's auc:0.724066
## [47]:  train's auc:0.761015  test's auc:0.723962
```

```
## [48]:  train's auc:0.761129  test's auc:0.723531
## [49]:  train's auc:0.761143  test's auc:0.723716
## [50]:  train's auc:0.761004  test's auc:0.723484
## [51]:  train's auc:0.761246  test's auc:0.723645
## [52]:  train's auc:0.761236  test's auc:0.723534
## [53]:  train's auc:0.761442  test's auc:0.723663
## [54]:  train's auc:0.761506  test's auc:0.723684
## [55]:  train's auc:0.761774  test's auc:0.723692
## [56]:  train's auc:0.761849  test's auc:0.724009
## [57]:  train's auc:0.762022  test's auc:0.72405
## [58]:  train's auc:0.762051  test's auc:0.724009
## [59]:  train's auc:0.762443  test's auc:0.723898
## [60]:  train's auc:0.762628  test's auc:0.724355
## [61]:  train's auc:0.762542  test's auc:0.724294
## [62]:  train's auc:0.762476  test's auc:0.724228
## [63]:  train's auc:0.762419  test's auc:0.724189
## [64]:  train's auc:0.762445  test's auc:0.723954
## [65]:  train's auc:0.76296  test's auc:0.723848
## [66]:  train's auc:0.763167  test's auc:0.723927
## [67]:  train's auc:0.763356  test's auc:0.724003
## [68]:  train's auc:0.763578  test's auc:0.724157
## [69]:  train's auc:0.763947  test's auc:0.724056
## [70]:  train's auc:0.764063  test's auc:0.724043
## [71]:  train's auc:0.764267  test's auc:0.724083
## [72]:  train's auc:0.764681  test's auc:0.724255
## [73]:  train's auc:0.76485  test's auc:0.723989
## [74]:  train's auc:0.765049  test's auc:0.723913
## [75]:  train's auc:0.765106  test's auc:0.723813
## [76]:  train's auc:0.765336  test's auc:0.723708
## [77]:  train's auc:0.765518  test's auc:0.723808
## [78]:  train's auc:0.7661  test's auc:0.7237
## [79]:  train's auc:0.766277  test's auc:0.723686
## [80]:  train's auc:0.766749  test's auc:0.723776
## [81]:  train's auc:0.766979  test's auc:0.723523
## [82]:  train's auc:0.767257  test's auc:0.72316
## [83]:  train's auc:0.767439  test's auc:0.723127
## [84]:  train's auc:0.7676  test's auc:0.723181
## [85]:  train's auc:0.767666  test's auc:0.723
## [86]:  train's auc:0.767966  test's auc:0.72308
## [87]:  train's auc:0.768177  test's auc:0.723154
## [88]:  train's auc:0.768413  test's auc:0.723097
## [89]:  train's auc:0.768559  test's auc:0.723279
## [90]:  train's auc:0.768721  test's auc:0.723247
## [91]:  train's auc:0.768888  test's auc:0.723499
## [92]:  train's auc:0.769265  test's auc:0.72364
## [93]:  train's auc:0.769481  test's auc:0.723877
## [94]:  train's auc:0.76973  test's auc:0.724167
## [95]:  train's auc:0.770122  test's auc:0.72429
## [96]:  train's auc:0.770414  test's auc:0.724181
## [97]:  train's auc:0.770656  test's auc:0.724331
## [98]:  train's auc:0.770943  test's auc:0.724403
## [99]:  train's auc:0.771186  test's auc:0.724353
## [100]:  train's auc:0.771585  test's auc:0.724308
## [101]:  train's auc:0.771832  test's auc:0.724568
## [102]:  train's auc:0.771978  test's auc:0.724624
## [103]:  train's auc:0.772049  test's auc:0.724579
```

```
## [104]:  train's auc:0.77214  test's auc:0.724641
## [105]:  train's auc:0.772252  test's auc:0.724465
## [106]:  train's auc:0.772414  test's auc:0.724508
## [107]:  train's auc:0.772807  test's auc:0.724497
## [108]:  train's auc:0.772798  test's auc:0.724619
## [109]:  train's auc:0.773231  test's auc:0.724418
## [110]:  train's auc:0.773276  test's auc:0.72457
## [111]:  train's auc:0.773524  test's auc:0.724648
## [112]:  train's auc:0.77363  test's auc:0.724827
## [113]:  train's auc:0.773709  test's auc:0.724856
## [114]:  train's auc:0.773803  test's auc:0.724812
## [115]:  train's auc:0.773849  test's auc:0.724982
## [116]:  train's auc:0.773934  test's auc:0.725144
## [117]:  train's auc:0.774345  test's auc:0.725057
## [118]:  train's auc:0.774603  test's auc:0.725245
## [119]:  train's auc:0.774922  test's auc:0.725527
## [120]:  train's auc:0.775177  test's auc:0.725639
## [121]:  train's auc:0.77539  test's auc:0.725868
## [122]:  train's auc:0.775703  test's auc:0.726146
## [123]:  train's auc:0.775926  test's auc:0.726354
## [124]:  train's auc:0.776089  test's auc:0.726457
## [125]:  train's auc:0.776381  test's auc:0.72646
## [126]:  train's auc:0.776495  test's auc:0.726545
## [127]:  train's auc:0.776695  test's auc:0.726755
## [128]:  train's auc:0.776915  test's auc:0.726881
## [129]:  train's auc:0.777115  test's auc:0.726966
## [130]:  train's auc:0.777255  test's auc:0.727025
## [131]:  train's auc:0.777448  test's auc:0.726975
## [132]:  train's auc:0.777583  test's auc:0.727026
## [133]:  train's auc:0.777728  test's auc:0.726924
## [134]:  train's auc:0.778021  test's auc:0.726782
## [135]:  train's auc:0.778219  test's auc:0.726744
## [136]:  train's auc:0.77831  test's auc:0.726714
## [137]:  train's auc:0.778415  test's auc:0.726864
## [138]:  train's auc:0.778628  test's auc:0.726748
## [139]:  train's auc:0.778936  test's auc:0.726639
## [140]:  train's auc:0.778954  test's auc:0.726686
## [141]:  train's auc:0.779071  test's auc:0.726818
## [142]:  train's auc:0.779232  test's auc:0.727034
## [143]:  train's auc:0.779672  test's auc:0.727074
## [144]:  train's auc:0.779816  test's auc:0.727109
## [145]:  train's auc:0.780004  test's auc:0.72726
## [146]:  train's auc:0.780216  test's auc:0.727358
## [147]:  train's auc:0.780462  test's auc:0.727374
## [148]:  train's auc:0.780595  test's auc:0.727534
## [149]:  train's auc:0.780886  test's auc:0.727589
## [150]:  train's auc:0.781138  test's auc:0.727429
## [151]:  train's auc:0.781413  test's auc:0.727312
## [152]:  train's auc:0.781699  test's auc:0.727306
## [153]:  train's auc:0.781893  test's auc:0.727192
## [154]:  train's auc:0.782239  test's auc:0.727295
## [155]:  train's auc:0.782393  test's auc:0.727217
## [156]:  train's auc:0.78259  test's auc:0.727421
## [157]:  train's auc:0.782807  test's auc:0.72736
## [158]:  train's auc:0.78316  test's auc:0.727263
## [159]:  train's auc:0.783345  test's auc:0.727451
```

```
## [160]:  train's auc:0.783697  test's auc:0.727346
## [161]:  train's auc:0.783933  test's auc:0.727322
## [162]:  train's auc:0.784109  test's auc:0.727222
## [163]:  train's auc:0.784279  test's auc:0.72723
## [164]:  train's auc:0.784412  test's auc:0.727179
## [165]:  train's auc:0.784641  test's auc:0.727099
## [166]:  train's auc:0.784832  test's auc:0.727076
## [167]:  train's auc:0.785061  test's auc:0.727089
## [168]:  train's auc:0.785409  test's auc:0.727134
## [169]:  train's auc:0.785684  test's auc:0.727154
## [170]:  train's auc:0.785964  test's auc:0.727064
## [171]:  train's auc:0.786141  test's auc:0.727204
## [172]:  train's auc:0.786316  test's auc:0.72718
## [173]:  train's auc:0.786453  test's auc:0.727205
## [174]:  train's auc:0.786632  test's auc:0.727233
## [175]:  train's auc:0.78674  test's auc:0.727218
## [176]:  train's auc:0.786923  test's auc:0.72734
## [177]:  train's auc:0.787276  test's auc:0.727271
## [178]:  train's auc:0.78758  test's auc:0.727244
## [179]:  train's auc:0.787841  test's auc:0.727217
## [180]:  train's auc:0.788124  test's auc:0.727368
## [181]:  train's auc:0.78843  test's auc:0.727425
## [182]:  train's auc:0.788715  test's auc:0.727684
## [183]:  train's auc:0.788921  test's auc:0.727724
## [184]:  train's auc:0.789177  test's auc:0.727903
## [185]:  train's auc:0.789408  test's auc:0.728002
## [186]:  train's auc:0.789599  test's auc:0.727997
## [187]:  train's auc:0.789787  test's auc:0.727956
## [188]:  train's auc:0.790024  test's auc:0.72781
## [189]:  train's auc:0.790258  test's auc:0.727806
## [190]:  train's auc:0.790522  test's auc:0.727818
## [191]:  train's auc:0.790705  test's auc:0.727713
## [192]:  train's auc:0.790993  test's auc:0.727614
## [193]:  train's auc:0.791261  test's auc:0.727572
## [194]:  train's auc:0.791404  test's auc:0.727294
## [195]:  train's auc:0.791588  test's auc:0.72723
## [196]:  train's auc:0.791726  test's auc:0.727218
## [197]:  train's auc:0.791996  test's auc:0.727145
## [198]:  train's auc:0.792348  test's auc:0.727154
## [199]:  train's auc:0.792557  test's auc:0.727186
## [200]:  train's auc:0.792746  test's auc:0.72719
```

```
## [1] "AUC: 0.728002022437408"
```

```
## [1] "Le score Gini pour le modèle LGMBoost : 0.456004044874816"
```

Étape 5 : Application des prédictions

Enfin, nous avons préparé un autre jeu de données (to_predict) pour la prédiction, en suivant la même procédure de prétraitement que pour les données de test. Nous avons ensuite utilisé notre modèle pour prédire les probabilités de sinistre pour ces données, ce qui peut être utilisé pour des décisions ultérieures dans le processus d'assurance.

Nous avons préparé un fichier de soumission comprenant les probabilités estimées de sinistre pour chaque bâtiment. La soumission de ce fichier sur le site du Data Challenge ENS a permis de convertir nos prédictions en un score de 0,3958.

Conclusion

Le modèle LGMBoost a montré des performances prometteuses avec une AUC de 0.7280, surpassant légèrement les résultats de l'approche XGBoost. Le score Gini, reflétant cette amélioration, atteint 0.4560, signifiant que notre modèle LGMBoost a réussi à mieux classer les bâtiments à risque par rapport aux modèles précédents.

La progression à travers les différentes méthodes de boosting a illustré comment l'ajustement fin des hyperparamètres, l'attention portée à l'équilibrage du modèle et l'évaluation rigoureuse de la performance peuvent aboutir à des améliorations significatives. La préparation minutieuse des données, l'optimisation des paramètres, et l'entraînement vigilant avec suivi des mesures de performance ont été des étapes clés pour atteindre une prédiction plus précise.

Conclusion

Au terme de notre étude dans le cadre du Data Challenge ENS sur l'assurance des bâtiments, notre analyse détaillée a permis de mettre en évidence la performance de différents modèles prédictifs, en mettant l'accent sur l'utilisation d'XGBoost et de LightGBM. Bien que chaque modèle ait offert des insights précieux, c'est finalement le modèle XGBoost qui a été retenu pour sa capacité supérieure à généraliser, comme en témoigne le score plus élevé obtenu sur le site du challenge ENS. Cette décision s'appuie sur la robustesse et la finesse des prédictions du modèle XGBoost, qui a su tirer parti des nuances complexes des données.

Dans notre modèle XGBoost, les caractéristiques comme la superficie du bâtiment `superficief`, le temps assuré `EXPO`, l'année de construction estimée `ft_22_categ`, le taux de cambriolage `Taux_de_Cambriolage` et le nombre d'inondations `Nombre_d_innondations` sont révélées être les plus discriminantes. Cela suggère logiquement que des facteurs comme la taille du bâtiment, son âge, la fréquence des sinistres locaux et la criminalité environnante sont cruciaux dans l'évaluation des risques de sinistre. Ces éléments reflètent les impacts directs et indirects sur la probabilité de réclamations, mettant en lumière la complexité de la tarification et de la gestion des risques dans l'assurance non-vie.

Le processus de modélisation a été exigeant mais extrêmement enrichissant. Nous avons approfondi notre compréhension des algorithmes avancés comme XGBoost et LightGBM, ce qui a nécessité une documentation approfondie et une curiosité technique constante. L'aspect le plus marquant et instructif de notre travail a été le traitement des données. L'ajustement précis des hyperparamètres et la manipulation adéquate des variables ont considérablement amélioré la précision de nos prédictions. Ce soin dans la préparation des données a renforcé notre modèle, permettant une interprétation plus claire et une prédiction plus fiable.

L'incorporation de données externes s'est avérée plus compliquée que prévu. La recherche et l'intégration de données pertinentes sur des éléments tels que les catastrophes naturelles et la criminalité locale ont présenté des défis significatifs, notamment en termes de fiabilité et de compatibilité des données. Cela a souligné l'importance d'une approche rigoureuse dans la sélection et le traitement des informations externes.

Pour des améliorations futures, enrichir notre modèle avec un éventail plus large de données externes pourrait être bénéfique. L'intégration de données supplémentaires sur des facteurs environnementaux et socio-économiques pourrait permettre une évaluation encore plus précise des risques. Cela pourrait également aider à mieux capturer les dynamiques locales qui influencent la fréquence et la sévérité des sinistres.