Vehicle Insurance Fraud Detection

ActuarAI — Machine Learning & Déploiement Applicatif

Ethan Ada & Tom Cohen
Master 2 — IMAFA

Janvier 2026

Contents

1 Introduction

La fraude en assurance automobile représente un enjeu économique et opérationnel majeur pour les compagnies d'assurance. Selon les estimations du secteur, plusieurs milliards d'euros sont perdus chaque année en raison de déclarations frauduleuses, qu'il s'agisse de sinistres fictifs, de dommages volontairement exagérés ou de collusions entre assurés, réparateurs et témoins.

Au-delà de l'impact financier direct, la fraude dégrade la qualité du portefeuille assurantiel, entraîne une hausse généralisée des primes et mobilise des ressources humaines importantes au sein des cellules d'investigation. Dans ce contexte, l'automatisation partielle du processus de détection constitue un levier stratégique pour améliorer l'efficacité opérationnelle tout en maintenant un haut niveau de contrôle du risque.

L'objectif de ce projet est de concevoir une solution complète de *détection automatique de fraude* reposant sur des techniques avancées de Machine Learning appliquées à des données tabulaires hétérogènes. Le projet, baptisé **ActuarAI**, s'inscrit dans une démarche actuarielle moderne, combinant :

- une modélisation statistique rigoureuse du risque de fraude,
- une évaluation quantitative basée sur des métriques adaptées aux classes déséquilibrées,
- et une mise en production du modèle sous la forme d'une application web interactive.

D'un point de vue méthodologique, le problème est formulé comme une tâche de **classification binaire supervisée**, où l'objectif est d'estimer la probabilité conditionnelle :

$$P(Y = 1 \mid X = x)$$

avec $Y = 1$ indiquant une fraude et X représentant le vecteur des caractéristiques décrivant l'assuré, la police d'assurance, le véhicule et l'incident déclaré.

Le modèle ne vise pas à se substituer à l'expertise humaine, mais à fournir un **outil d'aide à la décision** permettant de prioriser les dossiers à fort risque et d'optimiser l'allocation des ressources d'investigation.

Les objectifs opérationnels et scientifiques de ce projet sont les suivants :

- Analyser la structure et la qualité du jeu de données assurantiel,
- Concevoir un pipeline de prétraitement robuste et reproductible,
- Comparer des modèles linéaires interprétables à des modèles non linéaires de type boosting,
- Étudier l'impact du seuil de décision dans un cadre orienté coût métier,
- Déployer la solution sous forme d'une application web accessible à des utilisateurs non techniques.

2 Présentation du jeu de données

Le jeu de données utilisé est composé de **1000 observations** et de **40 variables explicatives** décrivant différentes dimensions du processus assurantiel. Ces variables peuvent être regroupées en quatre grandes catégories :

- **Informations contractuelles** : type de couverture, franchise, plafond de garantie, ancienneté du contrat, prime annuelle, limite parapluie (*umbrella limit*),

- **Profil de l'assuré** : âge, sexe, niveau d'éducation, profession, statut marital, loisirs déclarés,
- **Caractéristiques de l'incident** : date, localisation, gravité, type de collision, nombre de véhicules impliqués, présence de témoins, dommages corporels et matériels,
- **Informations sur le véhicule** : marque, modèle et année de mise en circulation.

La variable cible est définie par :

$$\texttt{fraud\_reported} \in \{0, 1\}$$

où 1 indique un sinistre frauduleux et 0 un sinistre légitime.

2.1 Distribution de la variable cible et déséquilibre de classes

L'analyse exploratoire met en évidence un déséquilibre de classes marqué :

- Environ 75.3% des observations correspondent à des sinistres non frauduleux,
- Environ 24.7% des observations correspondent à des cas de fraude.

Ce déséquilibre a une implication directe sur le choix des métriques d'évaluation. Une mesure telle que l'accuracy devient peu informative, car un classifieur trivial prédisant systématiquement la classe majoritaire atteindrait déjà une performance apparente élevée.

Dans ce contexte, les métriques basées sur les probabilités de classement, telles que la **ROC-AUC** et la **Precision-Recall AUC**, sont privilégiées car elles permettent d'évaluer la capacité du modèle à discriminer correctement les sinistres frauduleux indépendamment du seuil de décision choisi.

3 Prétraitement et ingénierie des variables

3.1 Suppression des variables identifiantes

Certaines variables présentent une unicité quasi parfaite et se comportent comme des identifiants techniques plutôt que comme de véritables caractéristiques explicatives. Leur présence peut conduire le modèle à mémoriser les observations du jeu d'entraînement, ce qui se traduit par un sur-apprentissage et une dégradation des performances en généralisation.

Les variables suivantes sont supprimées :

- `policy_number`,
- `incident_location`,
- `insured_zip`,
- `_c39`.

Ce choix est motivé par le principe actuariel selon lequel une variable explicative doit porter une information généralisable sur le risque, et non une information purement administrative.

3.2 Gestion des valeurs manquantes

Plusieurs variables catégorielles utilisent le symbole `"?"` pour représenter une information absente ou inconnue. Ces valeurs sont explicitement transformées en `NaN` afin de permettre une imputation statistique cohérente au sein du pipeline.

Les stratégies d'imputation retenues sont les suivantes :

- **Médiane** pour les variables numériques, afin de limiter l'influence des valeurs extrêmes,
- **Modalité la plus fréquente** pour les variables catégorielles, ce qui correspond à une hypothèse de neutralité en l'absence d'information.

Cette étape garantit la compatibilité du pipeline avec des données réelles potentiellement incomplètes lors du déploiement.

3.3 Ingénierie des variables temporelles

Les dates brutes présentent une forte cardinalité et peu de signification directe pour un algorithme d'apprentissage supervisé. Elles sont donc transformées en variables numériques interprétables :

- Année et mois de souscription du contrat,
- Année, mois et jour de survenue de l'incident,
- **Nombre de jours entre la souscription et le sinistre**.

Cette dernière variable constitue une proxy directe du comportement à risque : une survenue rapide d'un sinistre après la souscription est fréquemment associée à des schémas de fraude opportuniste.

4 Formulation du problème et pipeline de modélisation

Le problème est formulé comme une tâche de classification supervisée visant à estimer une fonction de décision :

$$f : R^p \to [0, 1]$$

où $f(x)$ représente la probabilité prédite qu'une déclaration x soit frauduleuse.

Le pipeline de traitement repose sur une séparation explicite des types de variables :

- Variables numériques : imputation par la médiane,
- Variables catégorielles : imputation suivie d'un encodage One-Hot.

L'ensemble est implémenté via un `ColumnTransformer`, garantissant la reproductibilité des transformations entre la phase d'entraînement et la phase d'inférence en production.

La séparation du jeu de données est réalisée selon un schéma 80/20 avec stratification sur la variable cible, afin de préserver la distribution du taux de fraude dans les ensembles d'apprentissage et de test.

5 Modèles entraînés

5.1 Régression logistique — Modèle de référence

La régression logistique constitue une baseline interprétable largement utilisée en actuariat pour la modélisation du risque. Elle repose sur l'hypothèse d'une relation log-linéaire entre les variables explicatives et le log-odds de la probabilité de fraude :

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$$

Ce modèle sert de point de comparaison afin d'évaluer le gain apporté par des méthodes non linéaires plus expressives.

5.2 HistGradientBoostingClassifier

Le modèle principal repose sur une approche de **boosting par gradient** adaptée aux données tabulaires. Le principe consiste à construire une somme additive de modèles faibles (arbres de décision) :

$$f(x) = \sum_{m=1}^{M} \gamma_m h_m(x)$$

où chaque arbre h_m est entraîné pour corriger les erreurs du modèle précédent selon le gradient de la fonction de perte.

L'implémentation par histogrammes permet une discrétisation efficace des variables continues, réduisant la complexité computationnelle et améliorant la stabilité numérique.

Étant donné que l'encodage One-Hot génère des matrices creuses, une transformation explicite en matrice dense est ajoutée dans le pipeline afin d'assurer la compatibilité avec l'algorithme.

6 Évaluation des performances et analyse décisionnelle

L'évaluation d'un modèle de détection de fraude ne peut pas se limiter à l'accuracy, notamment en raison du déséquilibre de classes (fraude minoritaire). Dans un contexte assurantiel, on privilégie des métriques de *discrimination* basées sur le classement des probabilités prédites.

Les performances sont évaluées à l'aide des métriques suivantes :

- **ROC-AUC** : mesure la capacité globale du modèle à classer un sinistre frauduleux audessus d'un sinistre non frauduleux,
- **Gini** : métrique actuarielle standard dérivée de la ROC-AUC, définie par :

$$\text{Gini} = 2 \times \text{AUC} - 1,$$

- **Precision-Recall AUC** : particulièrement informative en contexte de classes déséquilibrées (focus sur la classe fraude),
- **F1-score** : compromis entre précision et rappel pour un seuil donné,
- **Matrice de confusion** : lecture opérationnelle (FP/FN) utile pour estimer les impacts métier.

6.1 ROC-AUC et Gini : comparaison des modèles

Le **coefficient de Gini** est largement utilisé en scoring (crédit, fraude, assurance) car il fournit une mesure normalisée de la capacité de discrimination : un Gini proche de 0 indique un modèle équivalent au hasard, tandis qu'un Gini élevé traduit une séparation nette entre les deux classes.

Dans ce projet, deux modèles sont comparés : une **régression logistique** (baseline interprétable) et un modèle de **boosting** (HistGradientBoosting / XGBoost-like). Les résultats sur le jeu de test sont résumés ci-dessous :

Table 1: Comparaison des performances sur le jeu de test (discrimination)

Modèle	**ROC-AUC**	**Gini** $(2 \times AUC - 1)$	**PR-AUC**
Régression logistique	0.600	0.200	0.297
Boosting (HGB / XGBoost-like)	0.829	0.658	0.589

Ces résultats montrent une amélioration majeure de la discrimination : le modèle de boosting atteint un **Gini** ≈ 0.658, contre **Gini** ≈ 0.200 pour la régression logistique. Cette différence confirme que les relations non linéaires et les interactions entre variables (capturées par le boosting) sont déterminantes pour détecter des patterns de fraude complexes.

6.2 Optimisation du seuil et coût métier

Au-delà de la discrimination globale (AUC/Gini), la décision opérationnelle dépend du **seuil** appliqué à la probabilité de fraude. Dans un cadre assurantiel, les erreurs n'ont pas le même coût :

- **Faux négatif (FN)** : fraude non détectée $\Rightarrow$ paiement indu + perte financière potentiellement élevée,
- **Faux positif (FP)** : dossier légitime envoyé en enquête $\Rightarrow$ coût opérationnel + friction client.

Plutôt que d'utiliser un seuil arbitraire de 0.5, une analyse du F1-score en fonction du seuil est réalisée afin de trouver un compromis robuste entre précision et rappel.

Le seuil optimal (maximisant le F1-score) est obtenu pour :

$$\textbf{Seuil optimal} = 0.14$$

Ce choix favorise le **rappel** de la classe fraude (réduction des FN), ce qui correspond à une stratégie conservatrice souvent privilégiée en assurance : il est généralement préférable d'investiguer davantage de dossiers plutôt que de laisser passer des fraudes à fort coût.

7 Interprétabilité et robustesse du modèle

Dans un contexte assurantiel et de détection de fraude, la performance prédictive d'un modèle ne constitue pas un critère suffisant pour juger de sa qualité opérationnelle. Il est également essentiel de comprendre **quelles variables pilotent la décision** et d'évaluer la **stabilité du modèle** face à des perturbations des données ou à des changements dans les processus métier.

7.1 Analyse de l'importance des variables explicatives

Afin de quantifier la contribution réelle des variables explicatives à la capacité de discrimination du modèle, une analyse d'importance par *Permutation Importance* basée sur la baisse de la ROC-AUC a été menée. Cette approche consiste à permuter aléatoirement les valeurs d'une variable tout en conservant les autres inchangées, puis à mesurer la dégradation de la performance du modèle. Une baisse importante de l'AUC indique que la variable permutée joue un rôle central dans le classement des sinistres selon leur probabilité de fraude.

Les résultats montrent une forte concentration de l'information discriminante sur un nombre restreint de variables. La variable `incident_severity` se distingue nettement, représentant à elle seule plus de **70% de la baisse totale d'AUC observée**. Les variables `insured_hobbies` et `collision_type` apparaissent comme des facteurs secondaires, contribuant respectivement à environ **20%** et **1%** de l'importance globale. Les autres variables présentent des contributions marginales, suggérant qu'elles jouent principalement un rôle de raffinement local des décisions plutôt que de structuration globale du classement des risques.

Cette hiérarchie met en évidence le caractère fortement explicatif de certaines variables métier et souligne la capacité du modèle à exploiter des signaux comportementaux et contextuels pour améliorer la détection des schémas de fraude.

7.2 Test d'ablation et sensibilité du modèle

Afin d'évaluer la robustesse du modèle et de quantifier sa dépendance à la variable dominante `incident_severity`, un test d'ablation a été réalisé. Le modèle a été réentraîné après suppression complète de cette variable, puis comparé au modèle complet en termes de ROC-AUC et de coefficient de Gini.

Configuration du modèle	**ROC-AUC**	**Gini**
Modèle complet (toutes variables)	0.829	0.658
Modèle sans `incident_severity`	0.577	0.153

La suppression de cette variable entraîne une chute significative du pouvoir discriminant, avec un **delta de Gini** défini par :

$$\Delta\text{Gini} = 0.658 - 0.153 = \mathbf{0.505}.$$

Cette baisse marquée met en évidence une **forte sensibilité structurelle** du modèle à une variable métier clé. D'un point de vue opérationnel, cela constitue un point de vigilance majeur : toute évolution dans la manière dont la gravité des sinistres est déclarée, codifiée ou contrôlée pourrait entraîner une dégradation rapide des performances en production.

7.3 Implications opérationnelles et monitoring

Cette analyse souligne la nécessité de compléter la performance prédictive par une stratégie de **monitoring des variables critiques**. En particulier, la distribution de `incident_severity` devrait faire l'objet d'un suivi régulier afin de détecter d'éventuelles dérives de données (*data drift*) ou des changements de comportement déclaratif des assurés.

En pratique, ce type de surveillance peut s'appuyer sur des indicateurs tels que :

- la stabilité des distributions (tests de Kolmogorov-Smirnov ou PSI),
- l'évolution du ROC-AUC et du Gini sur des fenêtres temporelles glissantes,
- le suivi des taux de faux négatifs sur les dossiers audités.

Cette approche permet de garantir que le modèle conserve un niveau de fiabilité et de robustesse compatible avec des exigences opérationnelles et réglementaires élevées, tout en assurant une capacité d'adaptation face aux évolutions des processus métier et des comportements de fraude.

8 Déploiement applicatif — Streamlit

Le modèle final est déployé sous la forme d'une application web interactive développée avec **Streamlit**. Cette interface constitue une couche de valorisation métier permettant une exploitation directe du modèle par des profils non techniques, tels que des gestionnaires de sinistres ou des analystes fraude.

Lien public de l'application ActuarAI

https://actuarai-appgit-baghgz3pawqijznwg6vexs.streamlit.app/

L'application est hébergée sur **Streamlit Cloud**, garantissant une accessibilité immédiate via un navigateur web sans installation locale. Elle permet une interaction en temps réel avec le modèle de prédiction et constitue une démonstration fonctionnelle de la mise en production d'un pipeline complet de Machine Learning, depuis la préparation des données jusqu'à l'inférence opérationnelle.

8.1 Sélection du modèle déployé

Deux familles de modèles ont été évaluées en phase d'expérimentation : une **régression logistique** servant de modèle de référence interprétable, et un modèle de **boosting d'arbres de décision** (HistGradientBoosting / XGBoost-like) visant à capturer des relations non linéaires et des interactions complexes entre variables.

Bien que la régression logistique présente un avantage en termes de simplicité et d'interprétabilité, ses performances en discrimination se sont révélées significativement inférieures, avec un Gini proche de 0.20, contre un Gini supérieur à 0.65 pour le modèle de boosting. Cette différence traduit une capacité limitée du modèle linéaire à exploiter la richesse des signaux présents dans les données de sinistres.

Dans une perspective opérationnelle, où l'objectif principal est la **priorisation efficace des dossiers à investiguer**, la performance de classement prime sur la simplicité du modèle. En conséquence, seul le modèle de boosting a été retenu pour le déploiement applicatif, la régression logistique étant conservée comme **baseline analytique** pour les phases de validation, de comparaison et d'audit méthodologique.

8.2 Fonctionnalités principales

Les principales fonctionnalités incluent :

- **Saisie manuelle d'une déclaration de sinistre** pour une évaluation individuelle en temps réel avec affichage de la probabilité de fraude,
- **Chargement de fichiers CSV** pour le traitement batch de portefeuilles de sinistres,
- **Téléchargement des résultats enrichis** par la probabilité de fraude et le niveau de risque associé,
- **Visualisation du niveau de risque** sous forme de catégories opérationnelles : faible, modéré et élevé.

8.3 Architecture applicative

L'architecture de déploiement repose sur les composants suivants :

- Sérialisation du pipeline de prétraitement et du modèle via `joblib`,
- Chargement dynamique du modèle au démarrage de l'application,
- Interface utilisateur développée en Python avec Streamlit,
- Hébergement cloud via Streamlit Cloud, assurant la disponibilité publique et la reproductibilité des résultats.

Cette architecture garantit la cohérence entre la phase d'entraînement et la phase d'inférence, tout en offrant une solution légère, modulaire et facilement déployable dans un contexte académique ou industriel.

9 Limites et perspectives

Malgré les performances obtenues, plusieurs limites doivent être soulignées. Le jeu de données reste de taille modérée, ce qui peut restreindre la capacité de généralisation du modèle à des portefeuilles réels de grande dimension. De plus, certaines variables explicatives peuvent être sujettes à des biais déclaratifs.

Les perspectives d'amélioration incluent :

- L'intégration de méthodes d'explicabilité locale telles que SHAP pour fournir des justifications individuelles de décision,
- L'introduction de fonctions de coût asymétriques directement dans l'optimisation du seuil,
- La mise en place de mécanismes de détection de dérive des données,
- L'exploration de méthodes d'apprentissage semi-supervisé ou en ligne pour l'adaptation continue du modèle.

10 Conclusion

Ce projet propose une approche complète et opérationnelle de la détection de fraude en assurance automobile, articulant modélisation statistique, évaluation rigoureuse et déploiement applicatif.

Les résultats empiriques montrent que les modèles de boosting offrent une capacité de discrimination nettement supérieure aux approches linéaires traditionnelles, tout en restant compatibles avec des exigences d'exploitation métier.

L'intégration du modèle dans une application web transforme un prototype académique en un outil décisionnel exploitable, illustrant le potentiel des approches de Machine Learning dans la modernisation des processus actuariels et de gestion du risque.