# Data Challenge - Rapport

Marc KASPAR (M2 IASD)
Caio Rocha (M2 IASD)

Mars 2025

# Introduction

Prediction de la prime pure incendie en utilisant un modèle pour la Fréquence et un modèle pour le Coût moyen.

La Charge est obtenue en multipliant la fréquence, le coût moyen, et le nombre d'années depuis la souscription du contrat.

Données géographiques et Données spécifiques au contrat dont:

- L'activité de l'assuré
- Les indicateurs de souscription des garanties
- Le nombre de bâtiments, de salariés, et de sinistres déclarés lors de la souscription
- Les données de surface
- Les données de capitaux
- Les données liées à la prévention

# Travaux connexes

- Réseaux de neurones et GLM : modèles de référence [5] [1].
- Poisson : fréquence des sinistres (données de comptage) [6] [3].
- Gamma : sévérité des sinistres (données asymétriques).
- Prétraitement robuste (encodage, normalisation) essentiel [1].
- Forêts aléatoires et XGBoost : modèles fiables [4] [2].

# Analyse des données

- $X_{train}$ : 383610 éléments avec 374 features chacun.
- $X_{test}$ : 95852 élements
- 92 colonnes chaîne de caractère , 278 colonnes numeric, 2 colonnes mixtes
- 30.57% de NaN
- La variable cible "CHARGE" est fortement corrélée à "CM" ($\rho = 0.995$).

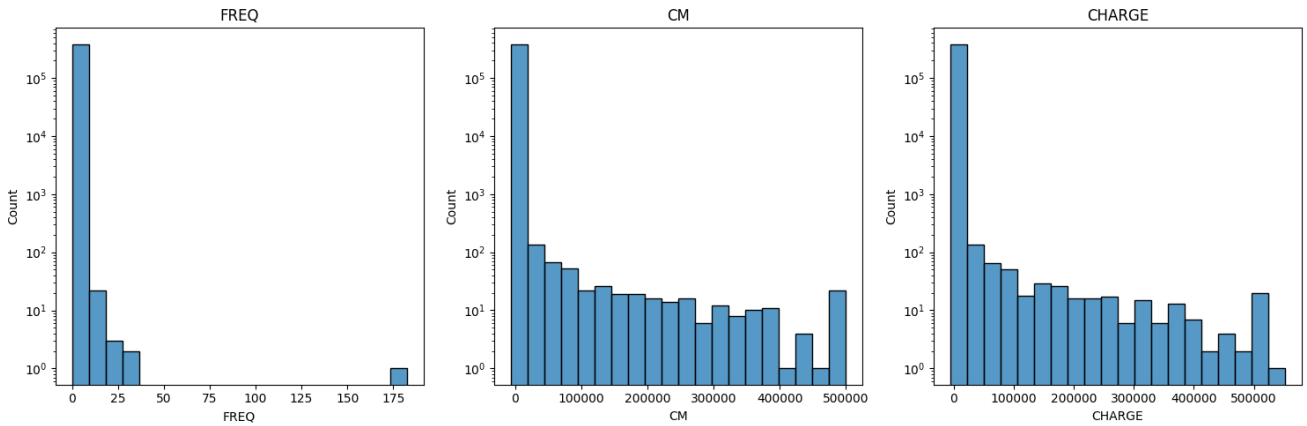



Figure: Distributions des variables étudiées.

# Analyse des données

- La majorité des colonnes sont catégoriques (dont certaines binaires).

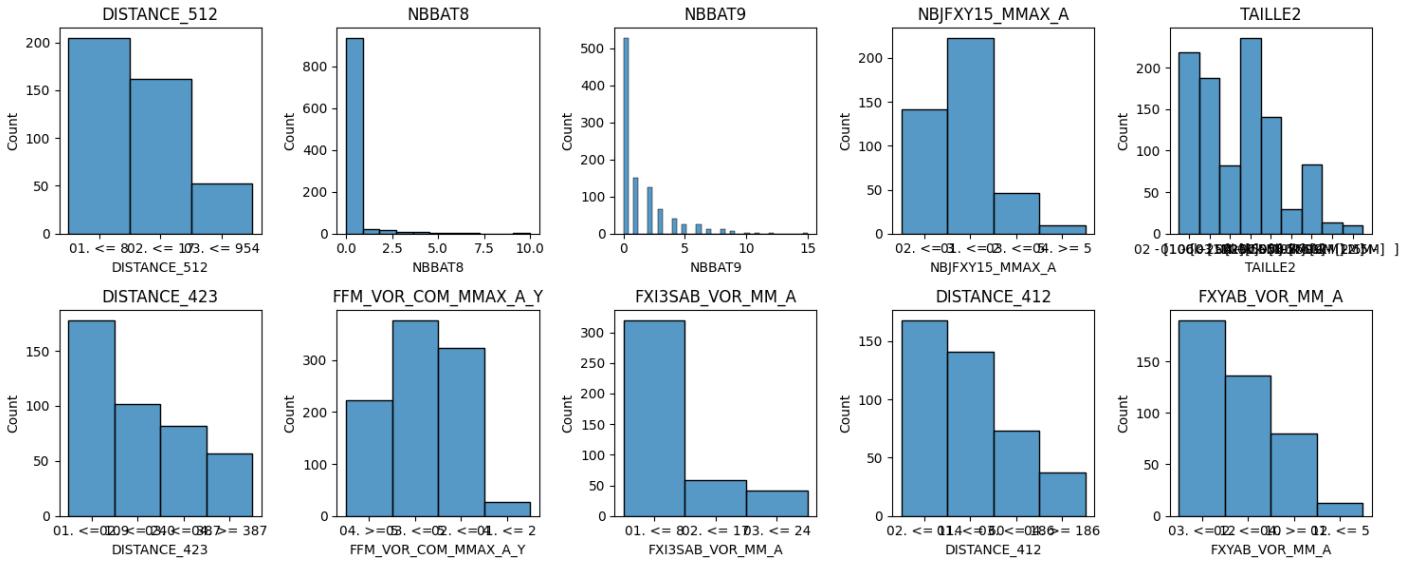
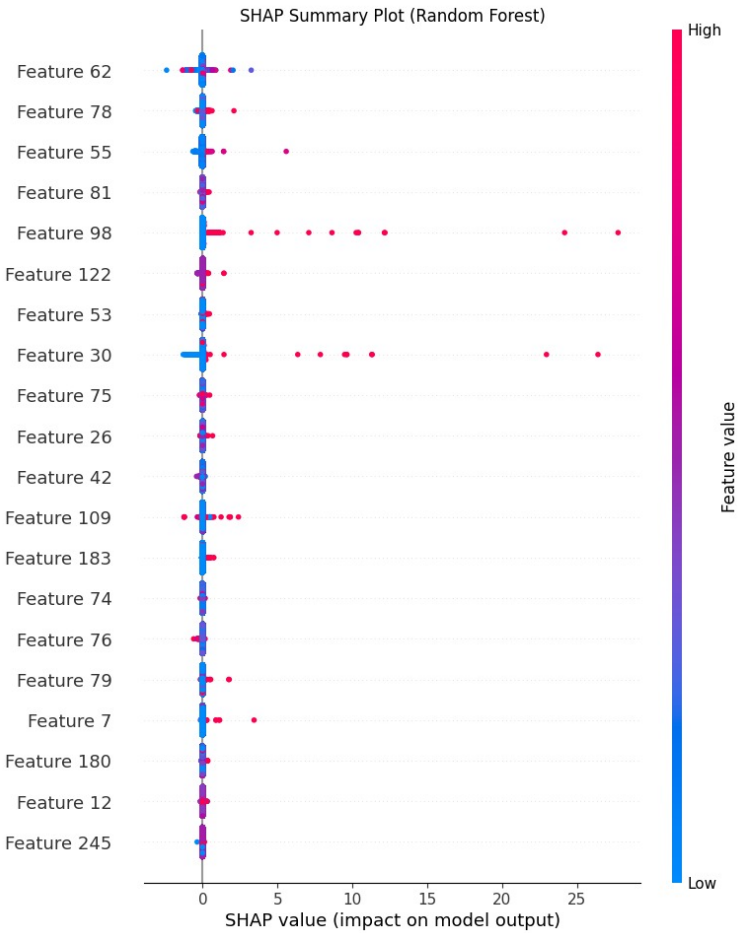
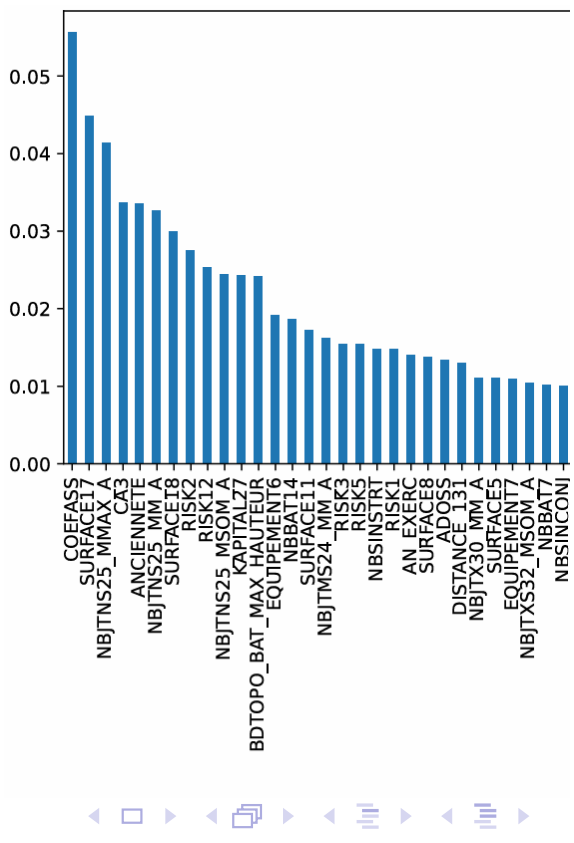



Figure: Distribution des données (10 attributs et 1000 observations aléatoirement sélectionnées pour la visualization).

# Feature Importance

- Analyse SHAP et Gini : importance des variables.
- Aucune variable dominante : information répartie sur toutes les caractéristiques.



# Prétraitement - Encodage

**Colonnes catégoriques :**

- Encodage ordinal pour colonnes binaires et catégoriques.
- Encodage One-Hot pour certaines colonnes spécifiques.

| | |
|---|---|
| 01 - [0 -250k] | 126652 |
| 02 - [250k-500k] | 79214 |
| 03 - [500k-750k] | 50552 |
| 05 - [1M - 1.5M] | 42991 |
| 04 - [750k- 1M] | 34424 |
| 06 - [1.5M - 2M] | 23073 |
| 07 - [2M - 3M] | 18786 |
| 08 - [3M - 4M] | 5113 |
| 09 - [4M - 5M] | 1777 |
| 10 - [5M - +[ | 1028 |
| Name: TAILLE1 | |

| | |
|---|---|
| ACT1 | 296563 |
| ACT5 | 53709 |
| ACT2 | 14359 |
| ACT8 | 5650 |
| ACT3 | 5335 |
| ACT9 | 4459 |
| ACT7 | 2201 |
| ACT6 | 1007 |
| ACT4 | 327 |
| Name: ACTIVIT2 | |

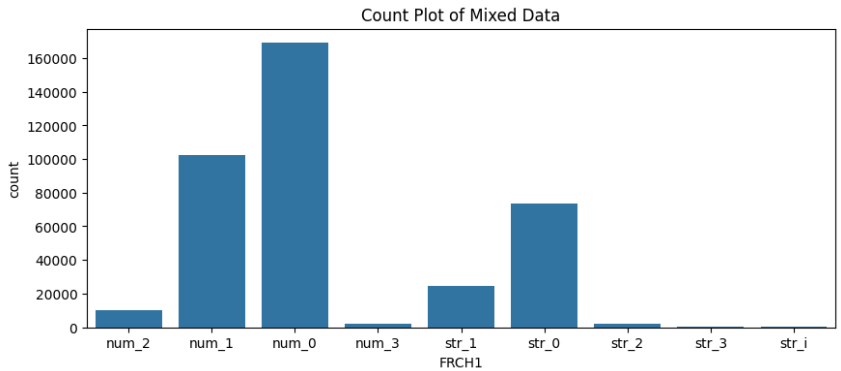
| | |
|---|---|
| 0 | 340077 |
| 01-10 | 11648 |
| 11-20 | 10769 |
| 21-30 | 7708 |
| 41-50 | 7057 |
| 31-40 | 6351 |
| Name: COEFASS | |

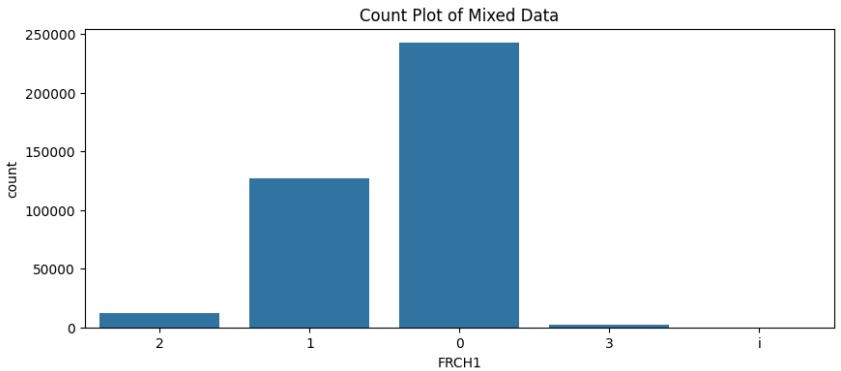
Figure: Examples d'attributs.

# Prétraitement - Colonnes Mixtes

**Colonnes mixtes :**

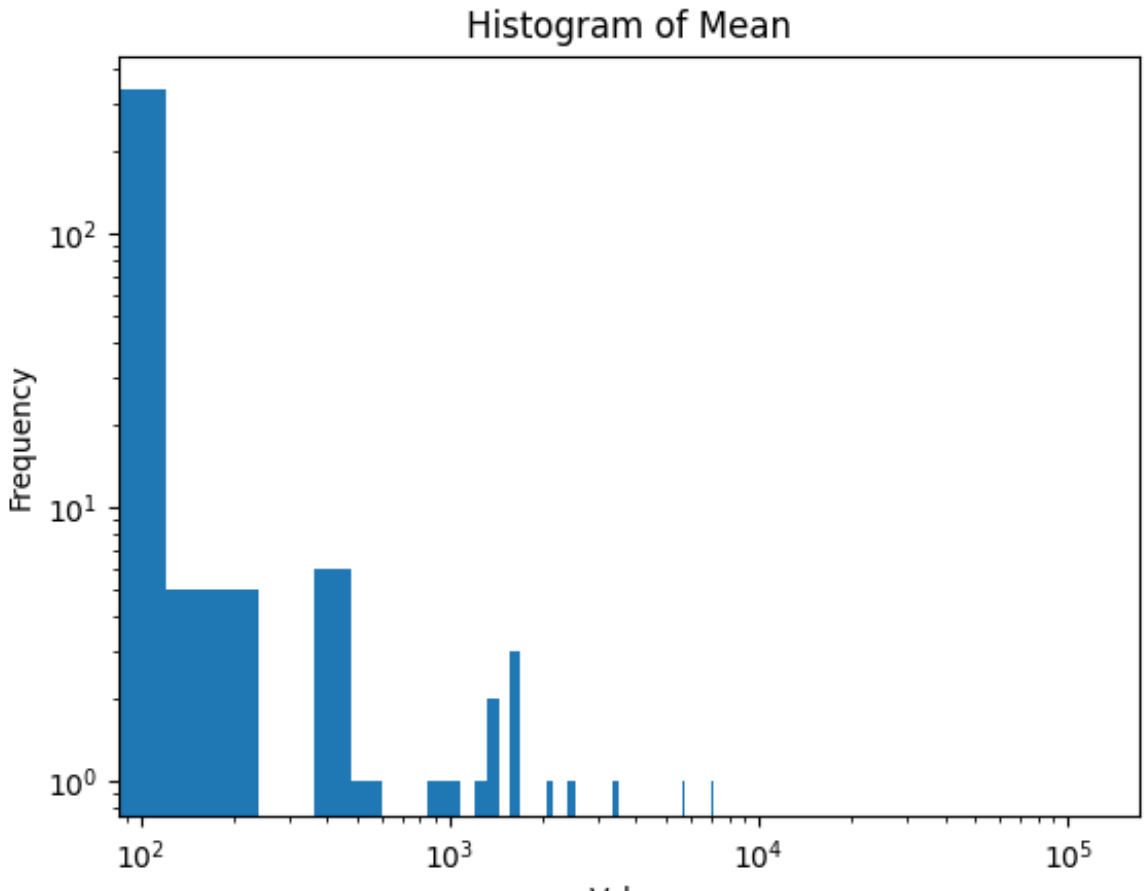
- Transformation des chaînes de caractères de nombres en données numériques (ex: "1" devient 1)
- Séparation en colonnes numériques et chaînes de caractères.
- One-hot encoding pour la colonne chaîne de caractère et imputation des valeurs manquantes avec la moyenne et normalization pour la colonne numérique.



# Prétraitement - Normalisation

- Imputation des NaNs par la moyenne.
- Normalisation par Z-Score (moyenne $= 0$, écart-type $= 1$).
- Suppression des colonnes contenant uniquement des NaNs après traitement.



# Taille des données

| | X_train | y_train | X_test |
|---|---|---|---|
| # observations | 383610 | 383610 | 95852 |
| # features (avant) | 372 | 4 | 372 |
| # features (après) | 399 | 4 | 399 |

Table: Dimensions des données.

# Modèles - Random Forest et XGBoost

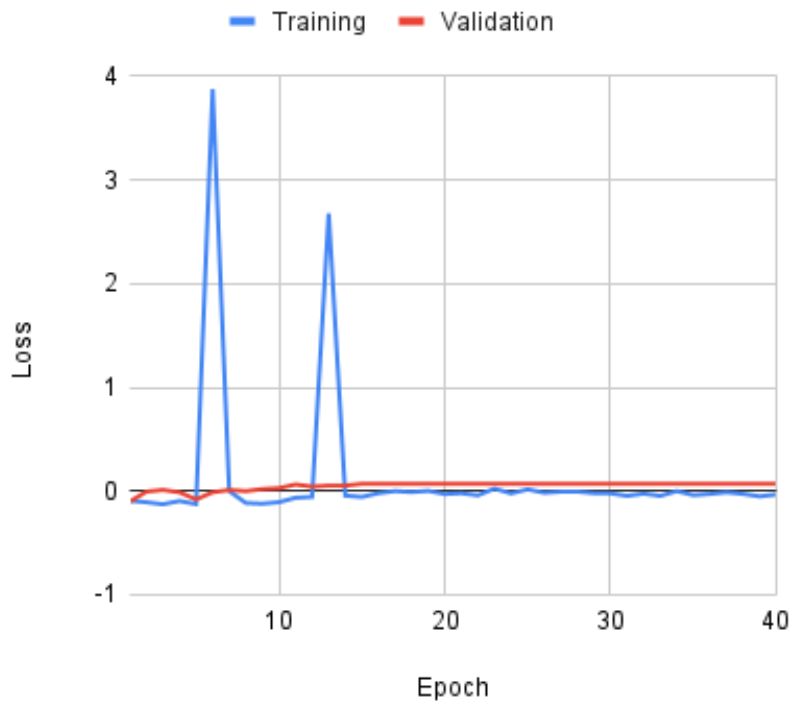
- Baseline : Random Forest pour prédire “FREQ” et XGBoost pour “CM”.
- Objectif : Évaluer l’impact du prétraitement sur la performance.
- Meilleure performance obtenue après encodage ordinal + One-Hot Encoding.

| Taille | Encoding | Normalization | Score |
|---|---|---|---|
| 383,610 | Ordinal + Binary + One hot | Yes | **5605.205184** |
| 383,610 | Count + One hot | Yes | 5606.166742 |
| 383,610 | Ordinal + Binary + One hot | No | 5607.787574 |
| 50,000 | Ordinal + Binary + One hot | Yes | 5725.741728 |
| 50,000 | Count + One hot | Yes | 5735.751156 |
| 50,000 | Ordinal + Binary + One hot | No | 5754.734910 |

# Modèles - Réseaux de neurones

- Implémentation de modèles avec fonction de perte Poisson ("FREQ") et Gamma ("CM").
- Convergence difficile à cause de l'implémentation de la perte Gamma.
- Résultats peu concluants.


(a) "FREQ"




(b) "Coût Moyen"

# Modèles - Régression linéaire

- Modèle linéaire simple testé avec régularisation (Lasso, Ridge).
- Actuellement en **5ème place** sur la plateforme !

| Model | Regularization ($\alpha$) | Frequence | | Mean Cost | | Charge | | Plateform |
|---|---|---|---|---|---|---|---|---|
| | | Train | Val | Train | Val | Train | Val | |
| Lin Reg | – | **0.2730** | 0.3710 | 5945.9390 | 6855.7070 | 49.75 | 100.72 | 5603,669 |
| Lasso | 0.01 | 0.2754 | **0.3679** | 5977.8804 | **6799.1216** | 7.32 | 8.32 | X |
| | 0.1 | 0.2754 | **0.3679** | 5982.2603 | 6801.0586 | **2.84** | **3.00** | X |
| | 1.0 | 0.2754 | **0.3679** | 5982.2603 | 6801.0586 | **2.84** | **3.00** | 5604,807 |
| | 10.0 | 0.2754 | **0.3679** | 5982.2603 | 6801.0586 | **2.84** | **3.00** | X |
| | 100.0 | 0.2754 | **0.3679** | 5982.2603 | 6801.0586 | **2.84** | **3.00** | X |
| Ridge | 0.01 | **0.2730** | 0.8984 | **5945.5728** | 11762.1914 | 50.05 | 2195278.75 | X |
| | 0.1 | **0.2730** | 0.4205 | 5945.8203 | 7240.1064 | 49.79 | 129755.34 | X |
| | 1.0 | **0.2730** | 0.3716 | 5945.9053 | 6860.6704 | 49.71 | 1777.75 | X |
| | 10.0 | **0.2730** | 0.3710 | 5945.9277 | 6855.0103 | 49.21 | 101.52 | X |
| | 100.0 | **0.2730** | 0.3707 | 5946.1274 | 6849.9077 | 46.06 | 93.57 | 5603,664 |

# Tentative d'amélioration

- Nous avons remarquer que lorsque la fréquence est 0, le coût moyen l'était également
- Nous avons essayé d'entrainer le coût moyen uniquement avec les éléments avec une fréquence non nulle.
- Cependant, les résultats étaient pires

# Travaux futurs

- Améliorer le prétraitement (encodage, gestion des NaNs).
- Optimiser les hyperparamètres de Random Forest et XGBoost.
- Reprendre les réseaux de neurones.

# Conclusion

- Les modèles linéaires offrent une bonne performance globale.
- Prétraitement des données = clé du succès.
- Potentiel d'amélioration avec les modèles de type réseau de neurones.


Piet de Jong and Gillian Z. Heller.
*Generalized Linear Models for Insurance Data*.
International Series on Actuarial Science. Cambridge University Press, 2008.


Muhammad Arief Fauzan and Hendri Murfi.
The accuracy of xgboost for insurance claim prediction.
*International Journal of Advances in Soft Computing and its Applications*, 10(2):159–171, 2018.
Publisher Copyright: © 2018, International Center for Scientific Research and Studies.


Joseph M. Hilbe.
*Poisson Regression*, page 35–73.
Cambridge University Press, 2014.


T. (Tim) Pijl.
A framework to forecast insurance claims.
Master's thesis, October 2017.


Anil Fernando Umar Isa Abdulkadir.

A deep learning model for insurance claims predictions.
*Journal on Artificial Intelligence*, 6(1):71–83, 2024.

Matthew D. Urban.
*Regression Models*.
Bookdown, 2023.