

**ISFA**

# **RAPPORT DE STAGE**

STAGE RÉALISÉ CHEZ

Nextuarial

Stagiaire : Mohamed El Hafed Ismail

Tuteur entreprise : Mohamed Amine Moutachakkir

Période : 2 Juin 2025 – 29 Juillet 2025

# Table des matières

<b>Introduction</b>	<b>2</b>
0.1 Introduction . . . . .	2
<b>Cadre théorique</b>	<b>3</b>
0.2 Cadre théorique du provisionnement . . . . .	3
0.2.1 Triangle de liquidation et notations . . . . .	3
0.2.2 Méthodes de provisionnement . . . . .	3
0.3 Machine Learning appliqué à la segmentation . . . . .	3
<b>1 Chatbot actuairel</b>	<b>4</b>
1.1 Contexte et objectifs . . . . .	4
1.2 Architecture technique . . . . .	4
1.3 Interface et fonctionnalités . . . . .	4
1.4 Fonctionnalités avancées . . . . .	5
1.5 Défis techniques et solutions . . . . .	5
1.6 Résultats et validation . . . . .	5
<b>2 Segmentation par machine learning</b>	<b>7</b>
2.1 Contexte et données . . . . .	7
2.2 Évolution méthodologique . . . . .	7
2.3 Validation et résultats . . . . .	8
2.4 Enseignements et limites . . . . .	9
<b>Conclusion</b>	<b>10</b>

# Introduction

## 0.1 Introduction

Ce rapport présente le travail effectué lors de mon stage de deux mois au sein de Nextuarial, du 2 juin au 29 juillet 2025. L'entreprise développe la plateforme ReAcT, qui implémente diverses méthodes de provisionnement actuariel (Chain Ladder, Mack, bootstrap, etc.) avec des fonctionnalités avancées .

Le stage s'est articulé autour de deux projets complémentaires visant à améliorer l'utilisation de cette plateforme par l'intelligence artificielle et l'analyse de données.

## Objectifs du stage

**Phase 1 : Chatbot actuariel** (4 semaines) Développer un chatbot basé sur GPT-4 permettant d'interroger les exports ReAcT en langage naturel, avec génération automatique de code Python et visualisations dans une interface Streamlit.

**Phase 2 : Segmentation par machine learning** (4 semaines) Créer une méthodologie de clustering pour segmenter les portefeuilles d'assurance et améliorer l'homogénéité des calculs de provisionnement.

## Contributions principales

La première phase a produit un prototype fonctionnel combinant interface conversationnelle, génération de code automatisée et système d'export multi-format. Le chatbot exploite efficacement les données structurées de ReAcT pour répondre aux questions métier.

La seconde phase a abouti à une méthodologie de segmentation validée statistiquement, avec un algorithme adapté aux contraintes actuarielles évitant la sur-simplification systémique. La segmentation optimale identifiée (12 clusters) offre le meilleur équilibre entre qualité statistique et pertinence métier.

Ces travaux illustrent l'application concrète de l'IA et du machine learning aux problématiques actuarielles, en alliant rigueur scientifique et besoins opérationnels.

# Cadre théorique

## 0.2 Cadre théorique du provisionnement

### 0.2.1 Triangle de liquidation et notations

On dispose d'un triangle de paiements  $\{X_{ij}\}_{i+j \leq n}$  où  $i$  désigne l'année d'origine,  $j$  le délai de développement, et  $C_{ij} = \sum_{k=0}^j X_{ik}$  le cumul associé. L'objectif est d'estimer la charge ultime  $S_i = C_{in}$  et la provision  $R_i = C_{in} - C_{i,n-i}$  [England and Verrall \[2002\]](#).

### 0.2.2 Méthodes de provisionnement

La méthode Chain Ladder postule que les ratios  $C_{i,j+1}/C_{ij}$  dépendent seulement de  $j$ . Les facteurs de développement  $\hat{f}_j = \frac{\sum_i C_{i,j+1}}{\sum_i C_{ij}}$  permettent de reconstruire la colonne ultime. Le modèle de Mack ajoute un cadre stochastique avec des hypothèses d'indépendance et de variance proportionnelle, permettant d'obtenir l'erreur quadratique moyenne (MSEP) nécessaire pour Solvabilité II [Mack \[1993\]](#), [Wüthrich and Merz \[2008\]](#).

Les méthodes Bornhuetter-Ferguson et Cape Cod combinent approches Chain Ladder et Loss Ratio pour lisser les extrêmes via la cadence de développement.

## 0.3 Machine Learning appliqué à la segmentation

Pour la segmentation des garanties, nous avons utilisé l'algorithme K-means après standardisation des variables. La qualité des partitions a été évaluée par le coefficient de silhouette [Rousseeuw \[1987\]](#) et l'indice de Davies-Bouldin [Davies and Bouldin \[1979\]](#).

Le choix du nombre de clusters  $k$  s'est fait via un score composite métier qui pénalise les segmentations trop simplistes ( $k=2$ ) et favorise une granularité actuariellement pertinente. Cette approche diffère des heuristiques standards (coude, gap statistic) qui ne convenaient pas aux contraintes actuarielles.

La validation privilégie la vérification que les groupes diffèrent sur des indicateurs clés (sévérité, délais) plutôt qu'une batterie de tests formels. L'objectif est d'obtenir des triangles plus homogènes pour améliorer la qualité du provisionnement.

# Chapitre 1

## Chatbot actuel

### 1.1 Contexte et objectifs

Dans le cadre de mon stage, j'ai développé une interface conversationnelle pour exploiter les outputs du logiciel ReAcT. L'objectif était de permettre aux actuaires d'interroger leurs données en langage naturel plutôt que de manipuler des fichiers Excel complexes.

La problématique identifiée était que les actuaires passent beaucoup de temps à extraire des informations spécifiques depuis ReAcT. J'ai donc créé un assistant IA conversationnel basé sur GPT-4.1 mini, développé en Python avec Streamlit, OpenAI API, Pandas et Plotly.

### 1.2 Architecture technique

Le projet est structuré autour de trois fichiers principaux : `core.py` pour la logique métier (chargement Excel, gestion conversations), `code.py` pour l'interaction OpenAI (génération et exécution sécurisée de code), et `app.py` pour l'interface Streamlit.

Le pipeline suit cette logique : ingestion des fichiers Excel → normalisation en DataFrames → traitement des requêtes via GPT-4.1 mini et GPT-4o → traitement de la réponse du LLM pour donner une réponse cohérente

### 1.3 Interface et fonctionnalités

J'ai choisi Streamlit pour sa rapidité de développement et son intégration Python native. L'interface comprend une zone de chat principal avec historique persistant et mémoire de conversation, affichage du code généré, rendu automatique des tableaux et visualisation Plotly.

La sidebar permet l'upload des fichiers Excel, la validation des données, les options d'export multiple (PDF, Excel, PowerPoint, Word, CSV, PNG) et un bouton reset pour

démarrer une nouvelle conversation.

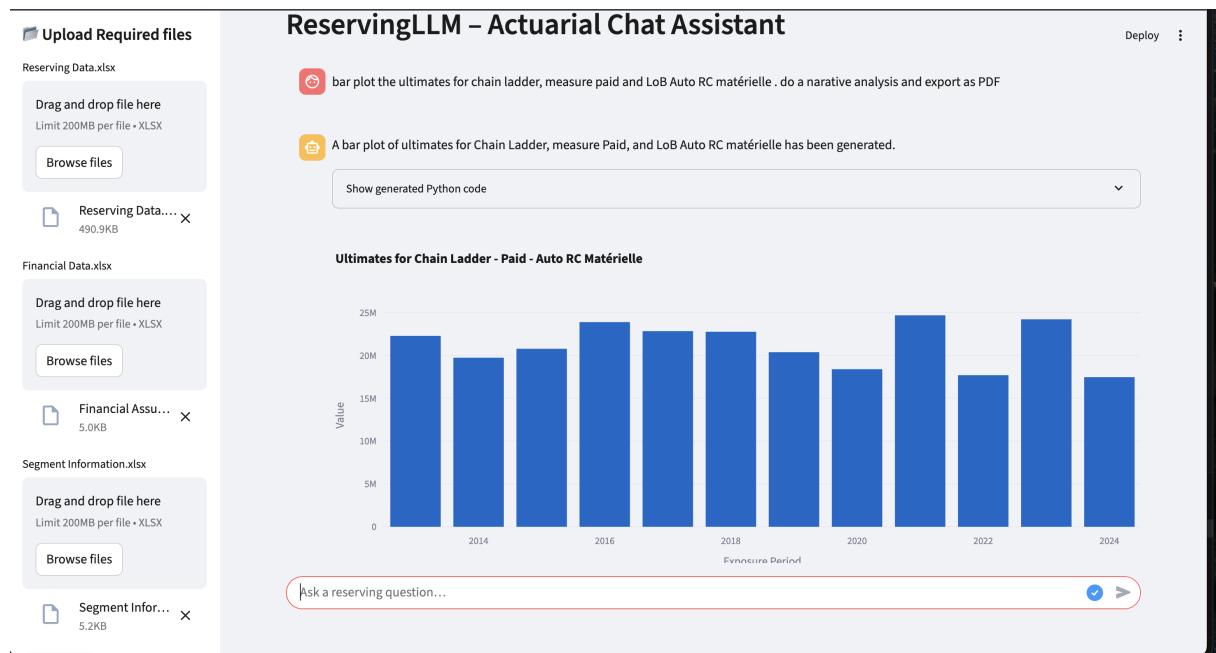


FIGURE 1.1 – Interface complète du chatbot avec options d’export

## 1.4 Fonctionnalités avancées

J'ai développé un système de prompts structuré qui guide GPT-4 pour générer du code Python conforme aux conventions et produire des réponses JSON formatées. Le système génère d'abord les calculs ou graphiques, puis crée un narratif actuariel combinant résultats quantitatifs et analyse qualitative.

Le système d'export permet l'export automatique sur commande textuelle et manuel via la sidebar, avec support de 6 formats différents, gestion des polices PDF et nettoyage automatique des fichiers temporaires.

## 1.5 Défis techniques et solutions

Les principaux défis concernaient la gestion des sessions Streamlit (résolu via `session.state`), l'intégration

Le développement itératif m'a permis de passer d'une interface basique à un système complet avec gestion d'erreurs robuste et optimisations UI/UX.

## 1.6 Résultats et validation

J'ai testé avec mon tuteur sur plusieurs réprises (deux fois par semaine au cours de l'avancement sur ce sujet) le système avec des requêtes simples (agrégations, filtres), com-

plexes (calculs d'ultimates, comparaisons multi-années) et des cas d'erreur. Le prototype développé apporte une réduction significative du temps d'analyse pour les actuaires et constitue une base solide pour un déploiement futur.

Cette phase m'a permis de maîtriser le développement d'applications Streamlit, l'intégration avec l'API OpenAI et la génération programmatique de documents multi-formats, tout en découvrant les défis du développement d'applications IA en entreprise.

# Chapitre 2

## Segmentation par machine learning

### 2.1 Contexte et données

L'objectif de cette phase était de segmenter les sinistres P&C en groupes homogènes pour améliorer la précision du provisionnement. Le principal défi consistait à identifier un jeu de données avec granularité sinistre-par-sinistre, multiplicité de LoB (15-25 lignes), dates complètes et volume suffisant.

Après avoir exploré plusieurs sources (fraudulent\\_claims, ausautoBI8999), j'ai adopté le package R SPLICE qui génère des données synthétiques multi-LoB réalistes. Le dataset final comprend 139,109 sinistres répartis sur 24 LoB segments couvrant Motor, Property, Liability et Specialty, avec paramètres individuels par LoB et variables essentielles (montants, délais, dates).

### 2.2 Évolution méthodologique

J'ai développé la méthodologie de clustering à travers plusieurs versions avec des apprennages successifs.

Les premières versions (v1-v2) utilisaient des données SPLICE enrichies avec 45 descripteurs (litigation rate, catastrophe exposure, comportement fraude), aboutissant à k=7 clusters avec des métriques correctes mais une complexité excessive.

La prise de conscience du problème (v3) est venue de l'analyse avec l'équipe : les variables polluaient les données et les rendaient trop complexes, masquant les vrais résultats actuariels attendus.

Pour la solution finale (v4), j'ai adopté une approche de purification en supprimant les variables artificielles et me concentrant sur les données SPLICE pures qui devraient être plus conformes à la réalité (16 descripteurs essentiels versus 45 précédemment). J'ai développé un algorithme de sélection business-oriented qui pénalise fortement k=2 (score -0.6), favorise la granularité (bonus +0.4 pour k10) et intègre la qualité statistique (silhouette,

séparation).

## 2.3 Validation et résultats

Pour valider l'impact actuariel, j'ai mis en place un back-test Chain-Ladder comparant trois stratégies : Baseline (24 triangles séparés), Clustered (triangles groupés selon notre segmentation) et All-Merged (triangle global unique). La performance était évaluée via RMSE, MAE et Bias.

La segmentation finale  $k=12$  présente une composition actuariellement cohérente regroupant les LoB par familles naturelles. Cette segmentation offre une réduction de variance (triangles plus homogènes), une granularité adaptée (12 segments vs 24 LoB individuelles), une cohérence métier respectant les familles de produits, et une opérabilité avec un nombre de triangles gérable.

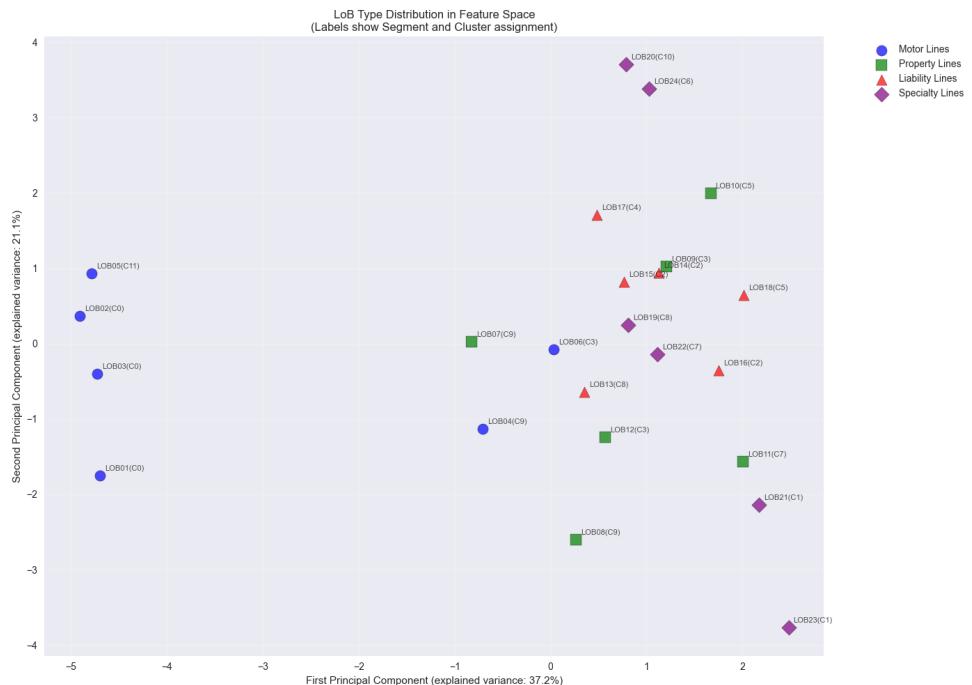


FIGURE 2.1 – Distribution des LoB dans l'espace des caractéristiques (projection PCA)

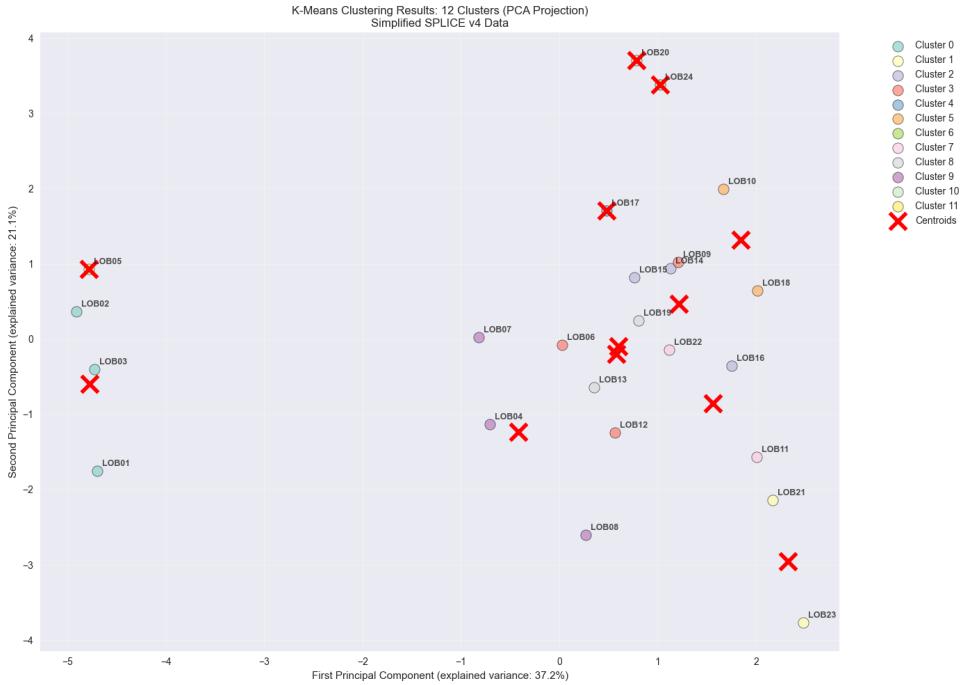


FIGURE 2.2 – Visualisation des 12 clusters finaux avec centroïdes

## 2.4 Enseignements et limites

Les enseignements méthodologiques majeurs incluent que la purification des données donne de meilleurs résultats que l'enrichissement artificiel, l'algorithme anti- $k=2$  évite la sur-simplification,  $k=12$  offre le compromis optimal entre précision et opérabilité, et le scoring business-oriented intègre les contraintes métier dans la sélection technique.

Les limites identifiées concernent le back-test incomplet (dataset SPLICE trop récent), certains clusters de taille variable, l'absence d'effets externes et l'utilisation d'un dataset fictif plutôt que de données réelles.

Les perspectives d'amélioration incluent l'exploration de méthodes alternatives (clustering hiérarchique, DBSCAN, mixture models), l'intégration d'embeddings non-linéaires (VAE, autoencoders), un back-test sophistiqué avec GLM bayésien et du stress-testing.

La segmentation  $k=12$  répond aux objectifs fixés : techniquement robuste (business score 0.709), actuariellement cohérente (respecte les familles de produits) et opérationnellement viable (12 triangles gérables vs 24 individuels). Cette méthodologie est directement transposable à tout portefeuille PC multi-LoB et constitue une base solide pour l'industrialisation de la segmentation actuarielle.

# Conclusion

## Conclusion

Ce stage de deux mois chez Nextuarial a permis de développer deux solutions innovantes combinant intelligence artificielle et expertise actuarielle : un chatbot d’interrogation des données de provisionnement et une méthodologie de segmentation de portefeuilles par machine learning.

## Réalisations principales

Le chatbot développé traduit avec succès les questions métier en analyses concrètes grâce à l’API OpenAI et une interface Streamlit. Il génère automatiquement du code Python, exécute les analyses et produit des visualisations pertinentes avec un système d’export multi-format. Cette solution démontre le potentiel de l’IA générative pour démocratiser l’accès aux données actuarielles complexes.

La méthodologie de segmentation a révélé l’importance de la purification des données sur leur enrichissement artificiel. L’algorithme développé évite la sur-simplification systémique ( $k=2$ ) en favorisant une granularité actuariellement pertinente, aboutissant à une segmentation optimale en 12 clusters validée par des métriques statistiques et des critères métier.

## Enseignements méthodologiques

L’approche itérative adoptée illustre l’importance de l’adaptation des techniques de machine learning aux spécificités actuarielles. La validation multi-critères (statistique et métier) s’avère essentielle pour l’évaluation des modèles en assurance. Le passage de 45 variables enrichies à 16 variables purifiées démontre que la simplicité bien maîtrisée surpassé souvent la complexité artificielle.

## Apports et perspectives

Ce stage a permis de maîtriser un écosystème technologique complet (Streamlit, OpenAI, scikit-learn, Pandas) tout en appliquant concrètement les connaissances académiques

en contexte industriel. Les méthodes de provisionnement étudiées à l'ISFA trouvent leur traduction pratique dans ReAcT, tandis que les techniques de clustering prennent sens dans la segmentation actuarielle.

Pour Nextuarial, ces travaux apportent des prototypes fonctionnels et une méthodologie éprouvée directement exploitables. Pour ma formation, cette expérience synthétise parfaitement théorie et pratique, confirmant l'importance de l'innovation technologique dans l'évolution des métiers de l'assurance.

Au-delà des livrables techniques, ce stage démontre que l'alliance entre IA et expertise actuarielle ouvre des perspectives considérables pour l'optimisation des processus. L'innovation technologique, pour être pertinente, doit s'ancrer dans une compréhension profonde des enjeux métier et s'appuyer sur une validation rigoureuse.

Cette mission s'inscrit dans l'évolution digitale de l'actuariat et pose les jalons d'une pratique professionnelle où la technologie amplifie l'expertise humaine sans la remplacer.

# Bibliographie

David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2) :224–227, 1979.

Peter D. England and Richard J. Verrall. *Stochastic Claims Reserving in General Insurance*. Cambridge University Press, 2002.

Thomas Mack. Distribution-free calculation of the standard error of chain ladder reserve estimates. *ASTIN Bulletin*, 23(2) :213–225, 1993.

Peter J. Rousseeuw. Silhouettes : a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20 :53–65, 1987.

Mario V. Wüthrich and Michael Merz. *Stochastic Claims Reserving Methods in Insurance*. John Wiley & Sons, 2008.