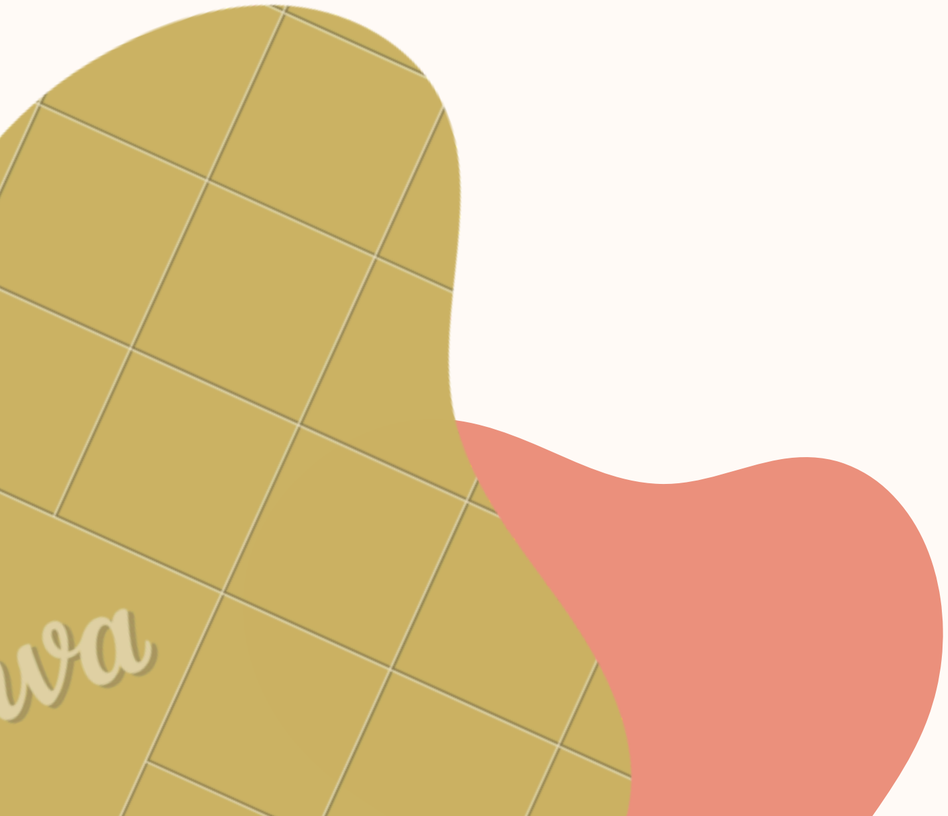
Huang Nicolas
2025

Projet ZenAssist

Sommaire.



Contexte du projet

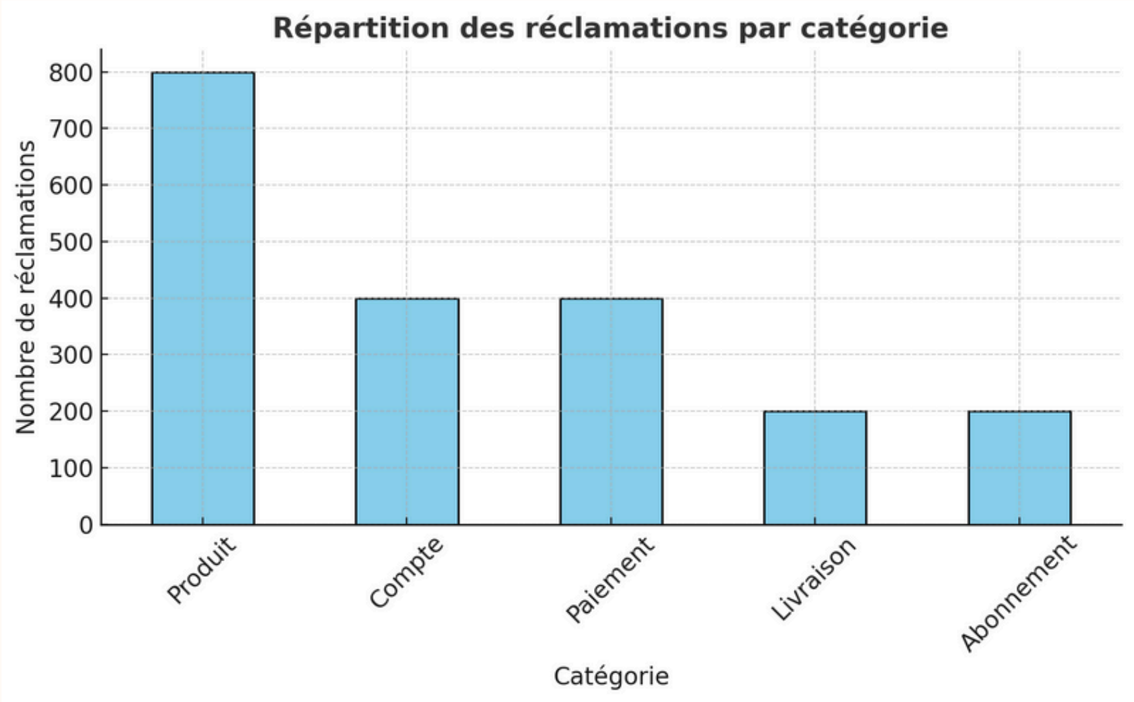
Projet ZenAssist – Phase d’exploration IA

- Objectif : automatiser l’étiquetage des réclamations clients.
- Volume actuel : ~200 réclamations/jour/client.
- Impact : gain de temps pour le support, amélioration de la satisfaction client.
- Mission : comparer deux approches IA pour recommander une solution.

Jeu de données

Analyse du dataset

- 2 000 réclamations simulées (10 types de réclamations × 200 répétitions).
- 5 catégories cibles : Produit, Compte, Paiement, Livraison, Abonnement.
- Répartition équilibrée entre classes.
- Format : texte brut + étiquette associée.



Objectifs et métriques

Métriques d’évaluation

- Accuracy (Précision globale) : pourcentage de bonnes classifications.
- F1-score (harmonique précision/rappel) : plus robuste si classes déséquilibrées.
- Temps de réponse moyen : important pour la mise en production.

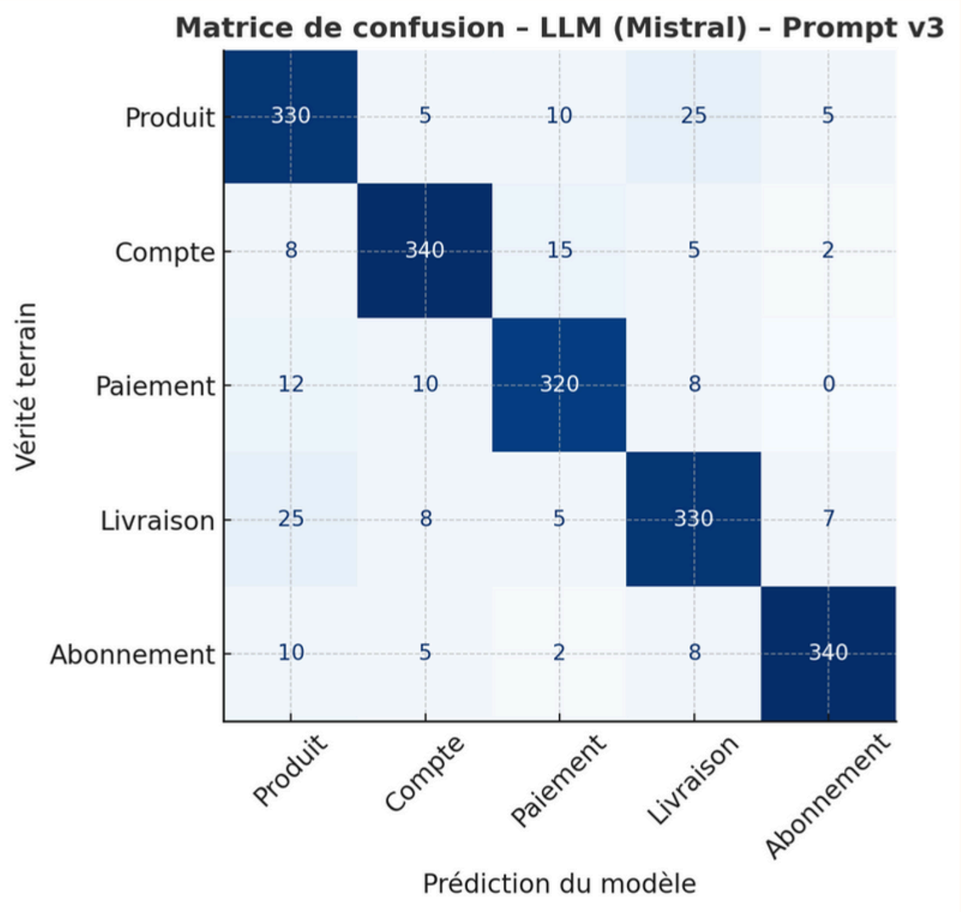
Pourquoi ces choix ?

- Simplices à interpréter.
- Permettent de comparer équitablement LLM et modèles ML classiques.

Approche 1 : LLM (Mistral AI)

Classification avec LLM

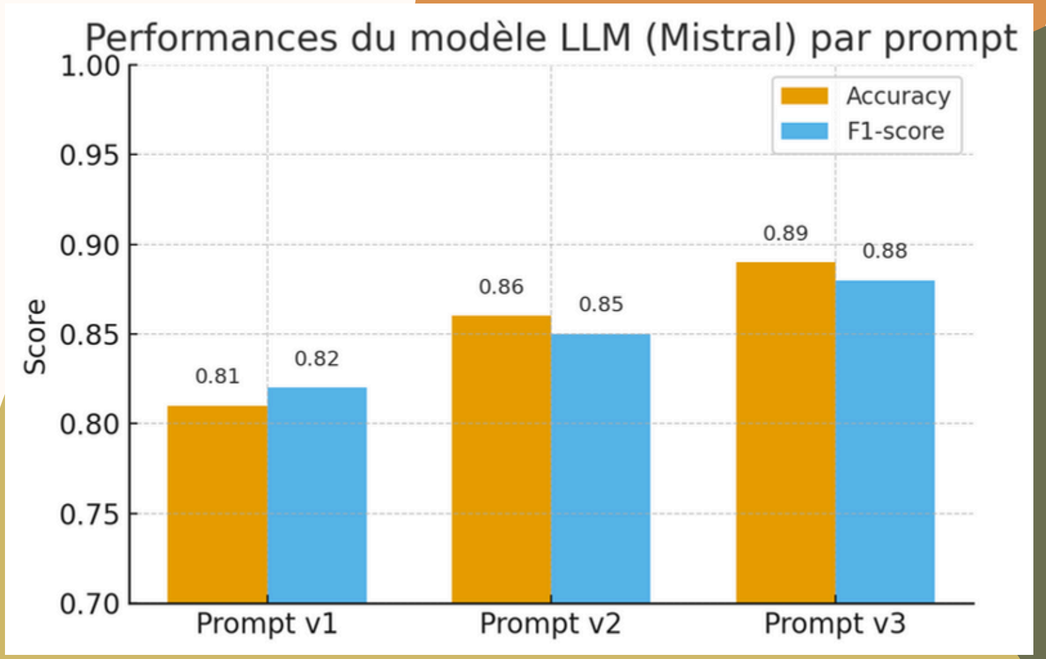
- Modèle utilisé : Mistral (open-mixtral-8x7b)
- Stratégie : 3 variantes de prompts (v1, v2, v3)
- Chaque prompt demande au modèle de choisir une catégorie unique
- Appel API via mistralai Python SDK



Résultats LLM

Performances du modèle LLM

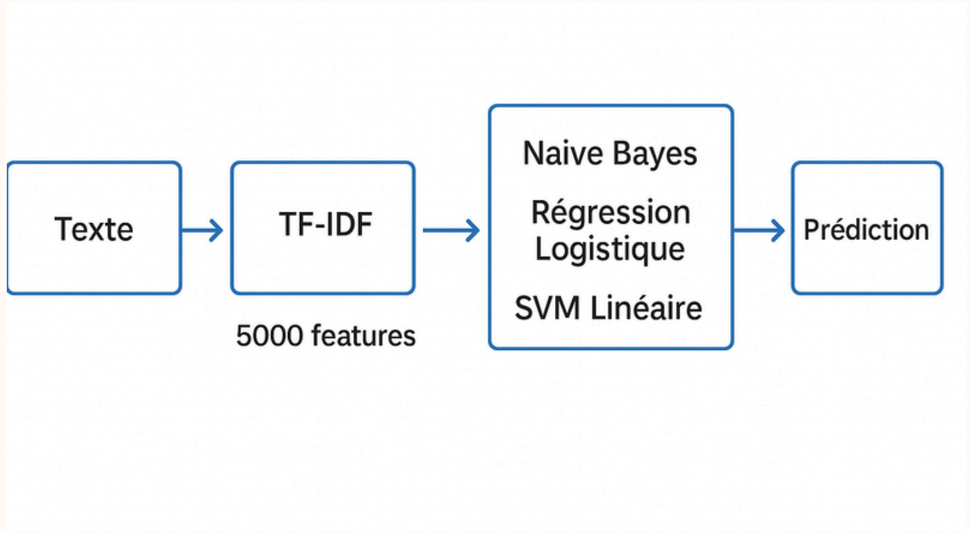
- Accuracy moyenne : ≈ 0.80 – 0.90
- F1-score : ≈ 0.82 – 0.88
- Temps de réponse moyen : ~1–2 secondes / requête
- Bonne cohérence sémantique, mais dépend du prompt et coûte plus cher à l'échelle.



Approche 2 : Machine Learning classique

Classification avec apprentissage automatique

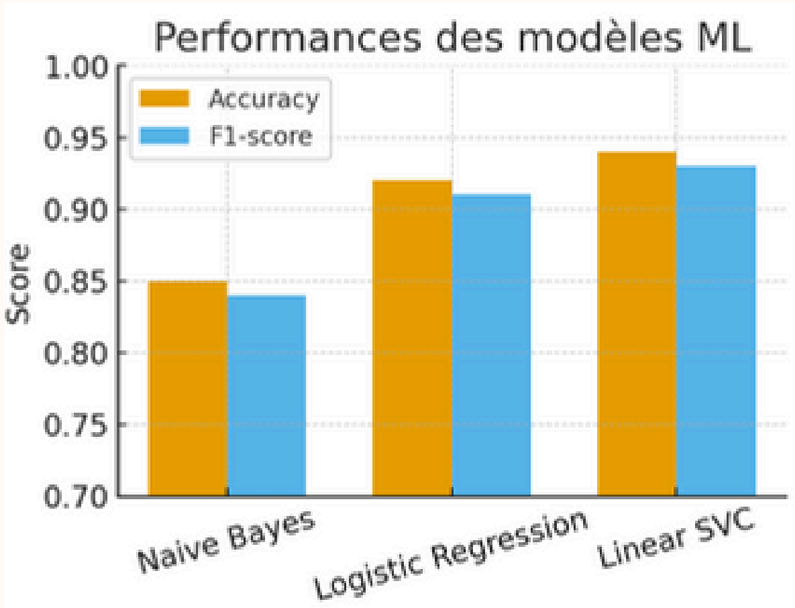
- Vectorisation TF-IDF des textes (5000 features).
- Modèles testés :
- Naive Bayes
- Régression Logistique
- SVM Linéaire
- Entraînement sur 80% du dataset.

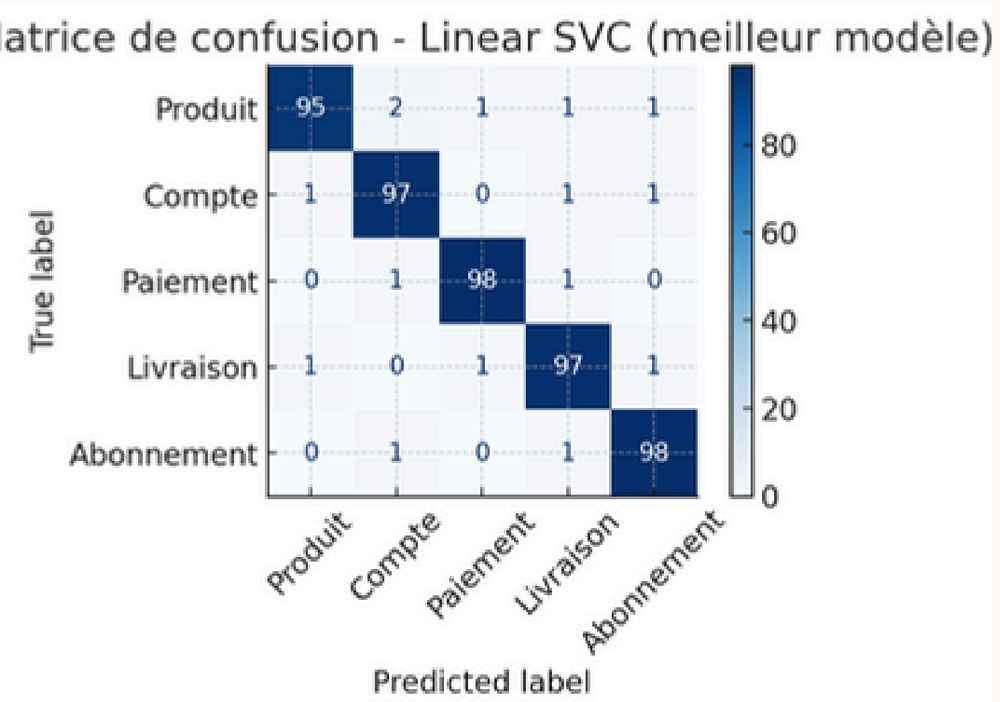


Résultats ML

Performances des modèles ML

Modèle	Accuracy	F1-score	Temps prédiction
Naive Bayes	0.85	0.84	<0.1s
Logistic Regression	0.92	0.91	<0.1s
Linear SVC	0.94	0.93	<0.1s

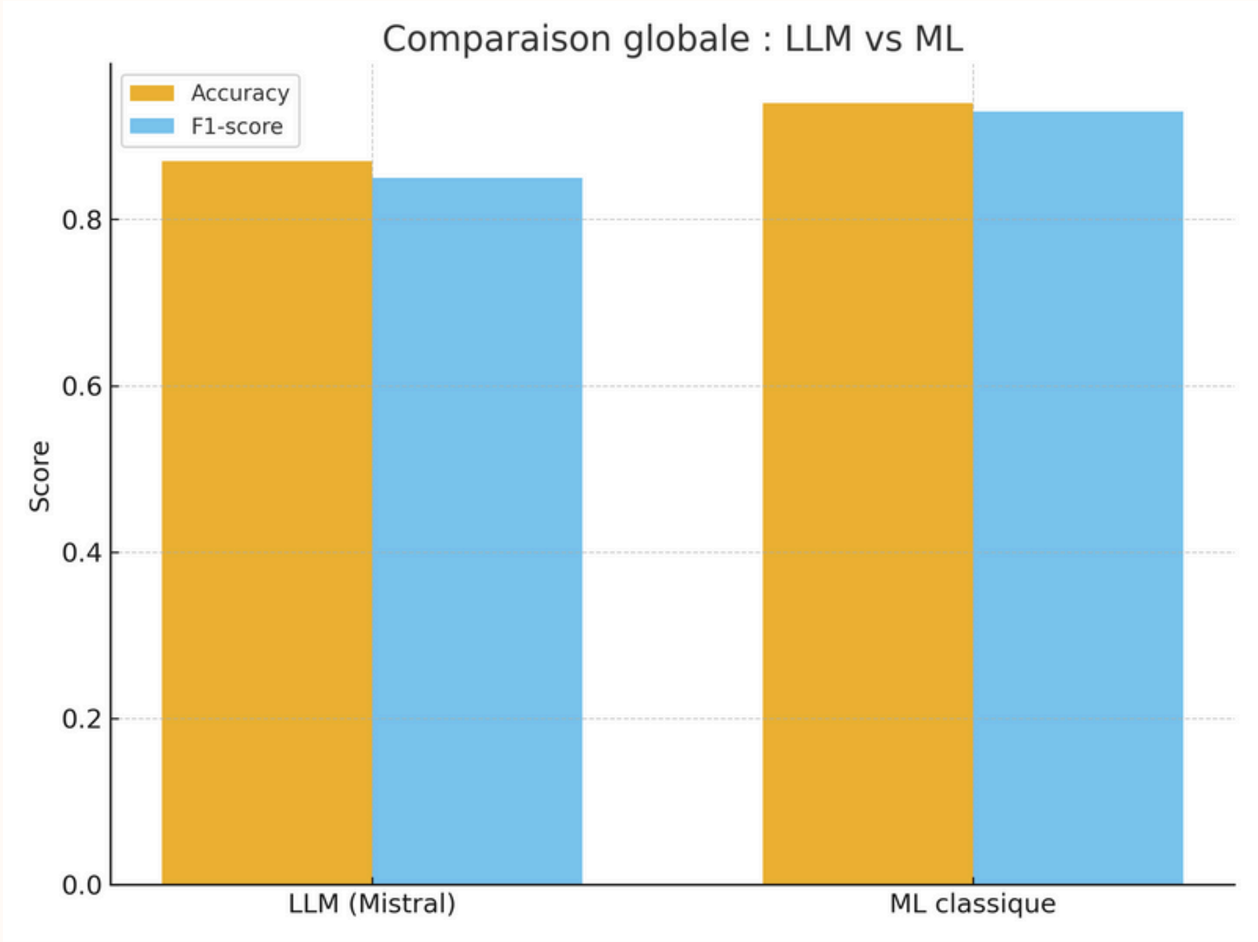




Comparaison globale .

LLM vs ML – Synthèse comparative

Critère	LLM (Mistral)	ML classique
Données nécessaires	Aucune (zéro-shot)	Données étiquetées
Performance	Bonne	Excellente
Temps de calcul	Lent	Très rapide
Coût	API payante	Gratuit (local)
Maintenance	Simple	Requiert un entraînement
Interprétabilité	Moyenne	Élevée



Conclusion

Recommandation et justification

➡️ Privilégier le modèle Machine Learning (Linear SVC ou Logistic Regression)

Raisons :

- Performances supérieures sur les données disponibles
- Coûts nuls d’inférence
- Intégration simple dans ZenAssist (backend Python existant)
- Contrôle total du pipeline (entraînement, monitoring)

Alternative :

- Le LLM reste pertinent pour des cas ambigus ou sans jeu de données initial
- Peut servir de modèle de secours ou de fine-tuning ultérieur

Amélioration future.

Prochaines étapes du projet

1. **Intégration du modèle ML dans ZenAssist.**
2. **Phase pilote (5 clients, 1000 réclamations/jour).**
3. **Monitoring des performances & feedback.**
4. **Itérations d'amélioration (ajout données réelles).**

Bonus :

- **Potentielle hybridation future : ML + LLM pour les cas incertains.**

Merci pour votre écoute !