Université de Caen Normandie

IUT Grand Ouest Normandie

Pôle de Caen

Bachelor Universitaire de Technologie

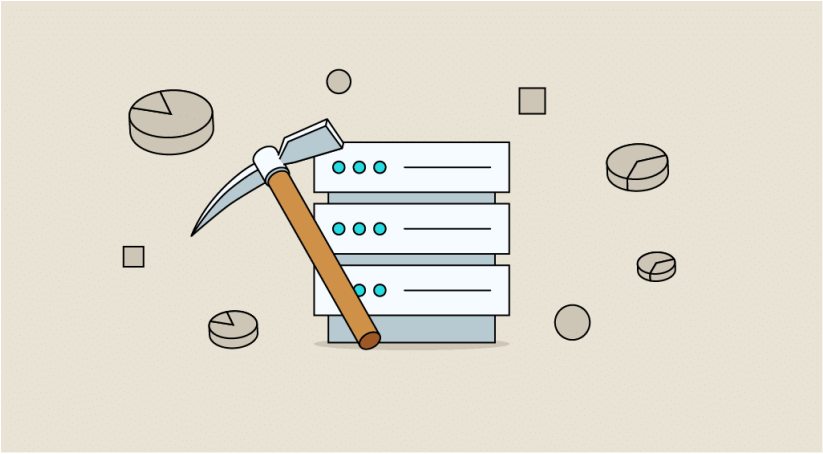
Science des Données

`Troisième année`

Ressource : Base de données NoSQL

---

analyse des commentaires utilisateurs et prédiction de la satisfaction client



Auteur

`Nomel DJAHOUA`

Année universitaire 2024-2025

SOMMAIRE

1 - Introduction

L'objectif de ce projet est d'analyser les commentaires des utilisateurs (variable "review_comment_message") en langue portugaise afin de comprendre les différents niveaux de satisfaction client. Nous avons classifié la satisfaction en quatre catégories : "très satisfait", "satisfait", "mécontent", et "insatisfait". Ce rapport présente les différentes étapes de traitement des données, la segmentation des commentaires et la mise en place d'un modèle de deep learning pour prédire la satisfaction des clients.

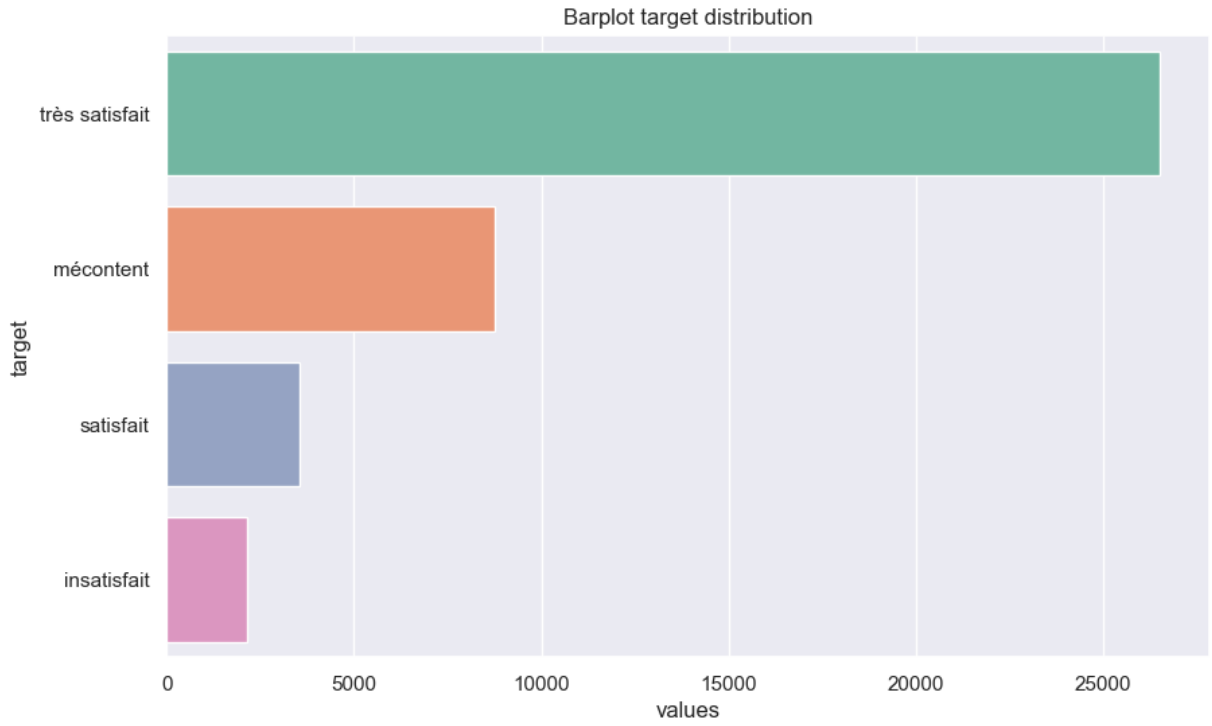
2 - Analyse du Jeu de Données

Le jeu de données contient 40 977 observations et 8 variables. Certaines colonnes, comme `review_comment_title`, contiennent un nombre important de valeurs manquantes (87 656 valeurs absentes).

Les catégories de la variable cible sont réparties comme suit :

— **Très satisfait :** 26 530 commentaires
— **Mécontent :** 8 745 commentaires
— **Satisfait :** 3 557 commentaires
— **Insatisfait :** 2 145 commentaires

```
target
très satisfait    26530
mécontent          8745
satisfait          3557
insatisfait        2145
Name: count, dtype: int64
```



Le graphique illustre une répartition statistiquement déséquilibrée des observations entre les différentes catégories de satisfaction. La catégorie "Très satisfait" domine largement, représentant la majorité des données, tandis que "Mécontent" suit avec un nombre nettement inférieur. Les catégories "Satisfait" et "Insatisfait" sont les moins fréquentes, avec des proportions significativement plus faibles.

3 - Traitement des données

Nettoyage des Données

Le processus de nettoyage des données a consisté à :

— **Standardisation des données textuelles :** Les commentaires textuels ont été uniformisés pour supprimer les variations inutiles (espaces superflus, casse, ponctuation inutile, etc.).

Tokenisation

La tokenisation a été utilisée pour transformer les textes en listes de mots, facilitant ainsi l'application des techniques de traitement naturel du langage (NLP). Ce processus décompose chaque commentaire en unités lexicales distinctes, appelées *tokens*.

Exemple de commentaires tokenisés :

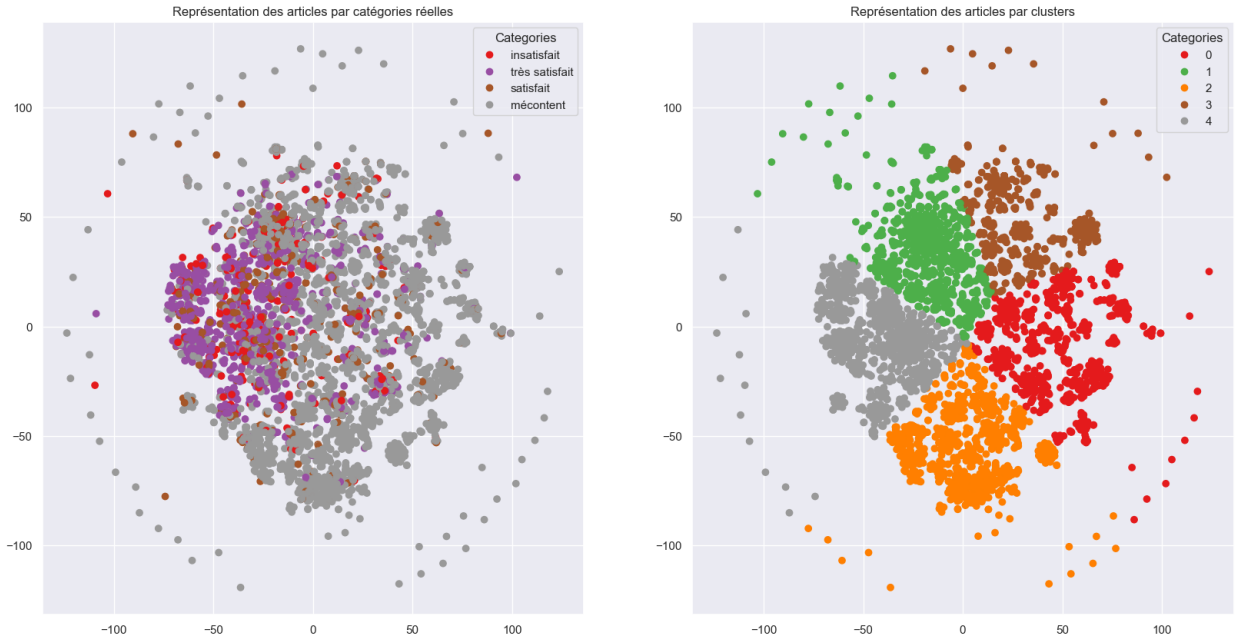
— **Texte original :** *"Recebi bem antes do prazo estipulado."*

— **Texte tokenisé :** `["recebi", "bem", "antes", "prazo", "estipulado"]`

3.1 Segmentation

Modèle de type Bags of Words

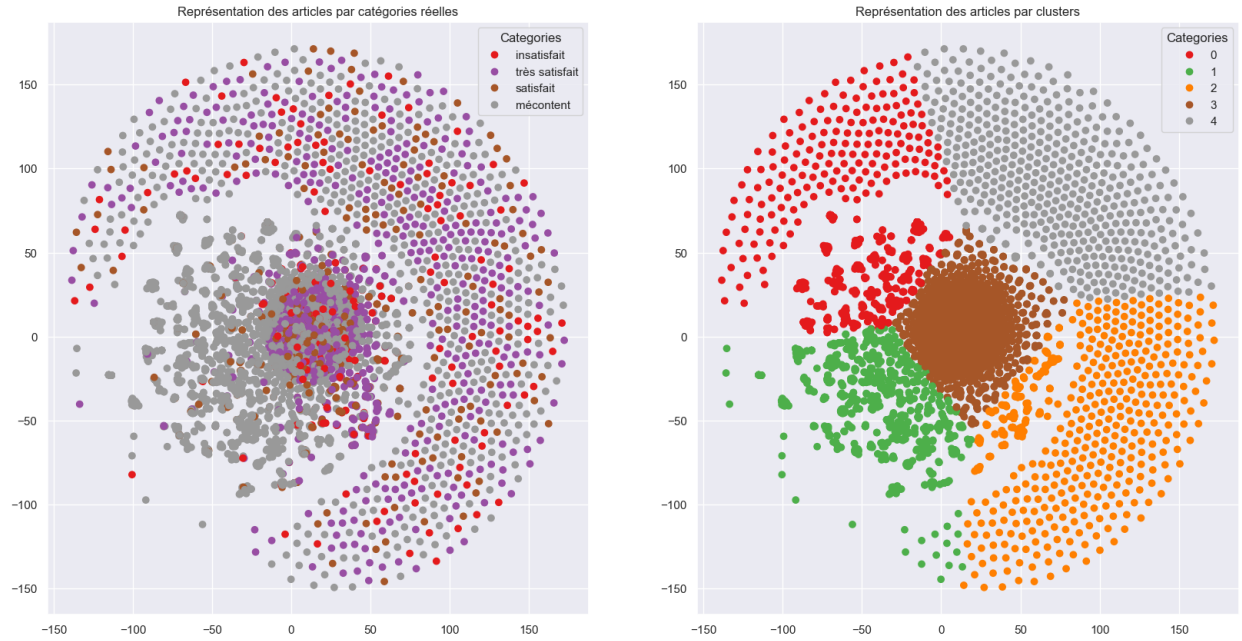
```
ARI :  0.0668
Time : 33.0
```



Les graphiques comparent les catégories réelles (gauche) et les clusters générés automatiquement (droite). À gauche, les points des différentes catégories sont mélangés, montrant une faible séparation visuelle entre elles. À droite, les clusters sont mieux définis et distincts, montrant que l'algorithme de regroupement a trouvé des groupes clairs. Cependant, il peut y avoir des désaccords entre les clusters et les catégories réelles, notamment dans les zones de mélange. Ces visualisations suggèrent que les données présentent des similarités importantes entre certaines catégories, rendant la classification plus complexe.

Modèle de type Bags of Words

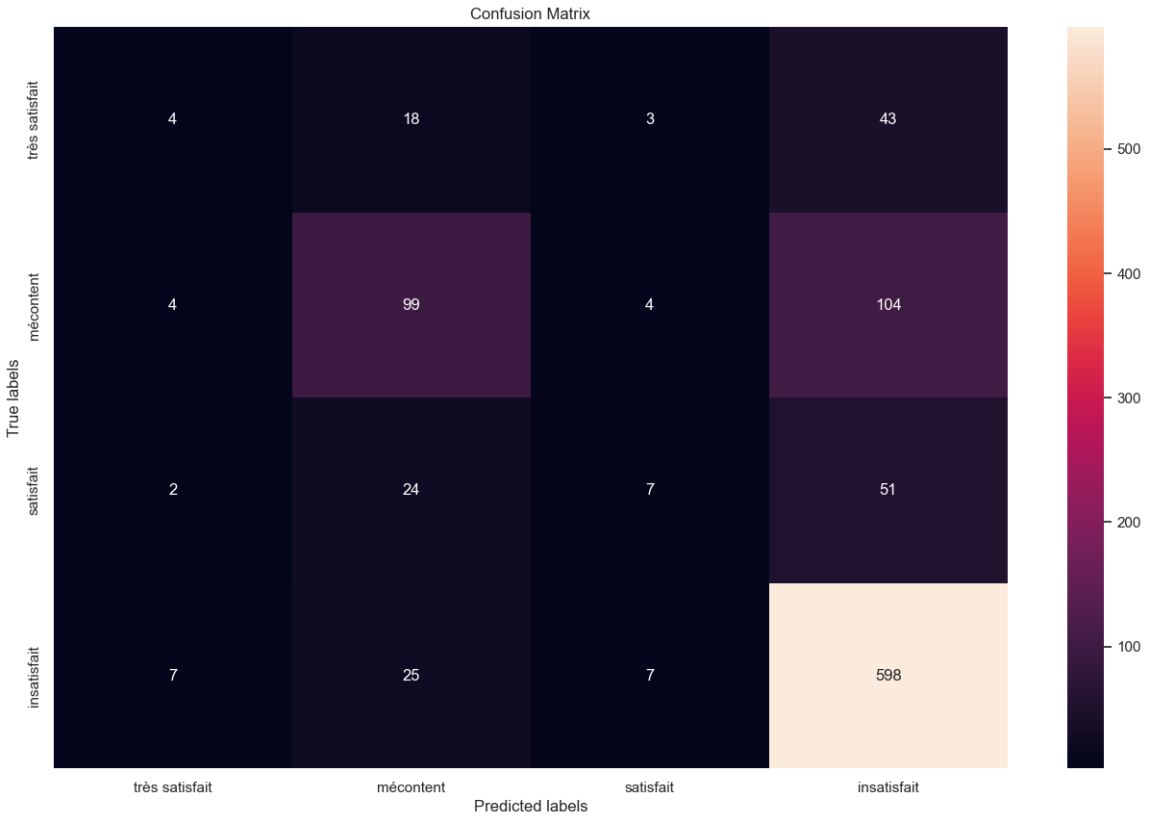
```
ARI :  0.055
Time : 33.0
```



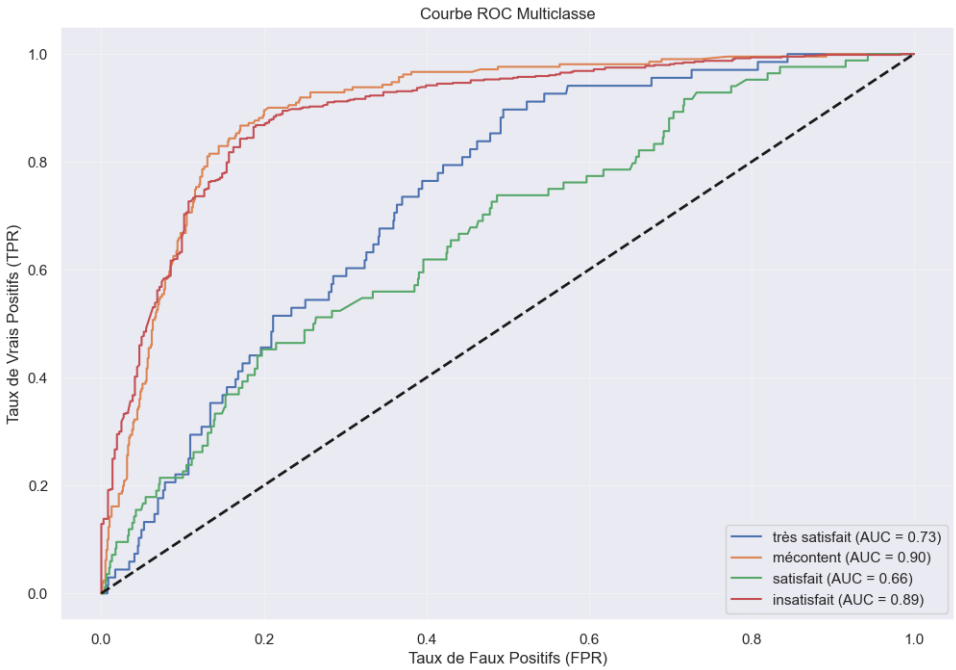
Les deux graphiques comparent la répartition des catégories réelles (à gauche) et les clusters générés automatiquement (à droite). À gauche, les points représentant les catégories réelles sont mélangés, montrant une faible séparation entre elles. À droite, les clusters générés sont nettement mieux définis, avec des regroupements distincts dans l'espace. Cela indique que l'algorithme de clustering a trouvé des structures dans les données, bien que leur correspondance exacte avec les catégories réelles reste à vérifier. Cette visualisation met en évidence la capacité des clusters à organiser les données de manière plus structurée.

3.2 Deep learning

Modèle de type Bags of Words

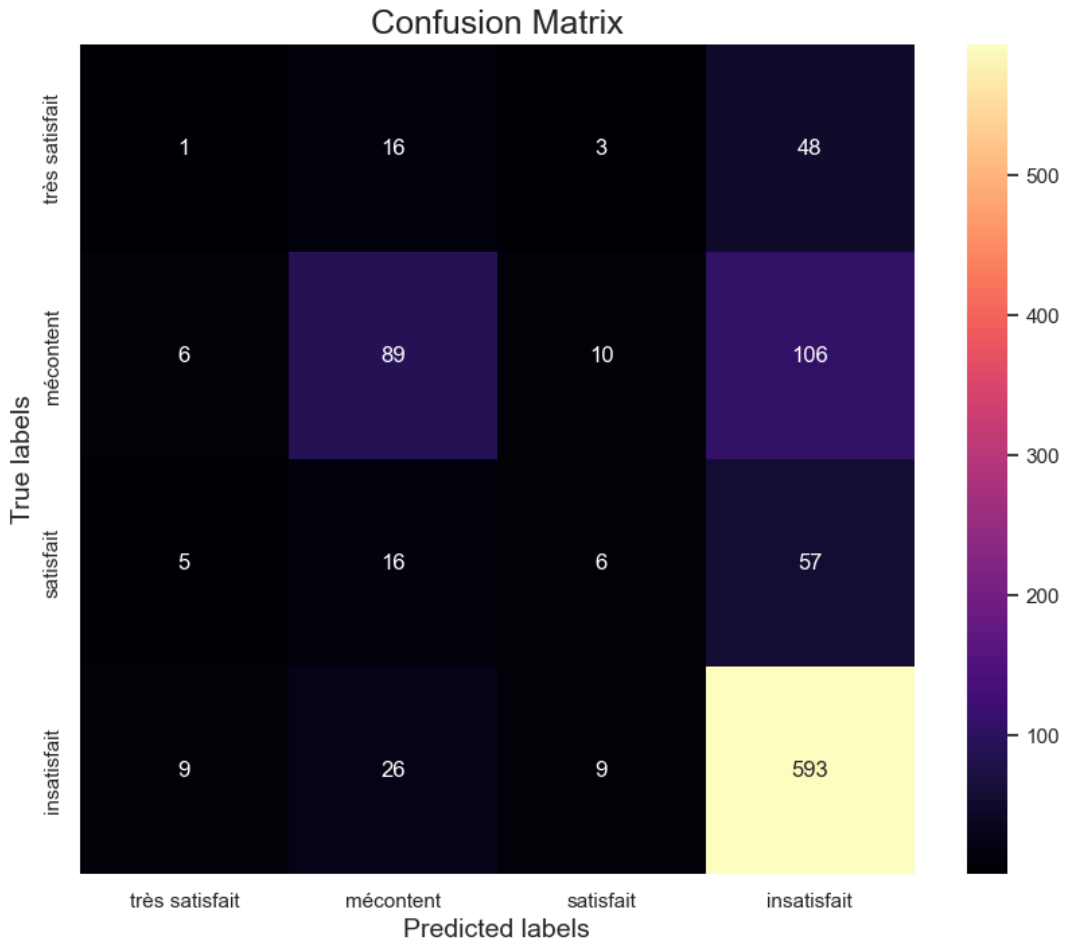


La matrice de confusion montre que la classe "insatisfait" est bien prédite avec 598 bonnes prédictions, mais il existe des confusions significatives pour les autres classes, comme "mécontent" et "très satisfait". Cela peut indiquer un déséquilibre des données ou des caractéristiques insuffisantes pour séparer correctement les classes.

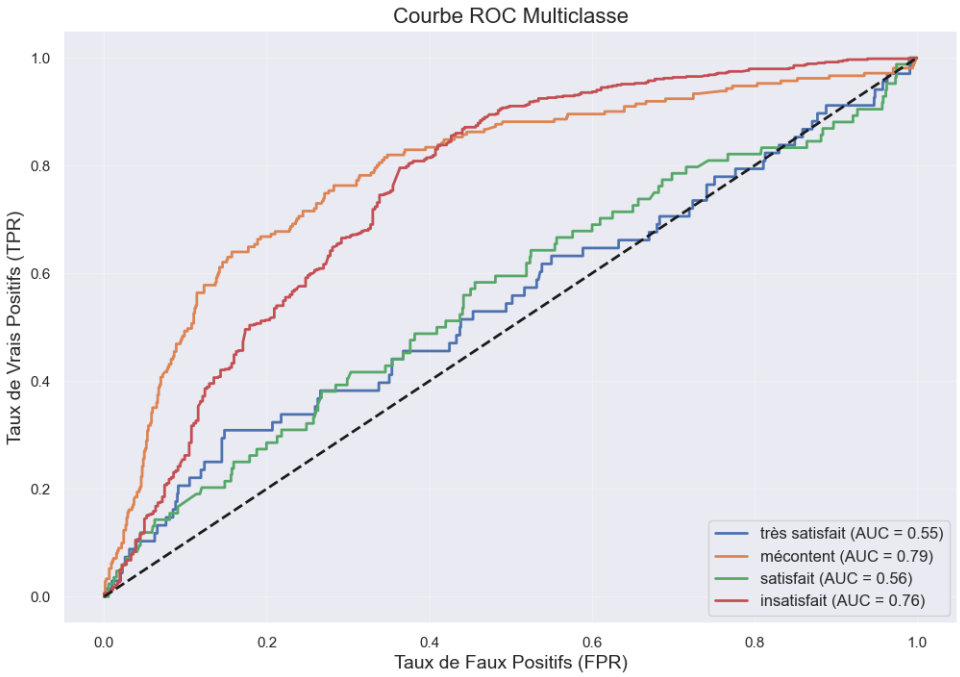


La courbe ROC multiclasse montre les performances du modèle pour chaque catégorie. Les valeurs AUC indiquent que le modèle est performant pour les classes "mécontent" (AUC = 0,90) et "insatisfait" (AUC = 0,89), tandis que les classes "très satisfait" (AUC = 0,73) et surtout "satisfait" (AUC = 0,66) sont moins bien discriminées. Cela suggère que le modèle distingue bien certaines classes, mais a plus de mal avec celles qui sont proches ou sous-représentées. Globalement, les performances sont correctes, mais il reste une marge d'amélioration pour les classes avec une faible AUC.

Modèle de type tf-idf



La matrice de confusion montre que les prédictions pour la classe "insatisfait" sont les plus précises avec 593 bonnes prédictions. Cependant, les classes "très satisfait", "mécontent" et "satisfait" ont des confusions significatives, notamment avec la classe "insatisfait". Cela peut indiquer un déséquilibre dans les données ou des caractéristiques insuffisantes pour discriminer certaines classes. Des approches comme la rééchantillonnage des données ou l'utilisation de modèles plus complexes pourraient améliorer les résultats.



La courbe ROC multiclasse montre que les performances du modèle varient selon les classes. Les classes "mécontent" (AUC = 0,79) et "insatisfait" (AUC = 0,76) ont les meilleures performances, indiquant une bonne capacité à les distinguer. En revanche, les classes "très satisfait" (AUC = 0,55) et "satisfait" (AUC = 0,56) ont des AUC proches du seuil de 0,5, reflétant une difficulté significative à les différencier. Cela suggère que le modèle a des lacunes pour prédire correctement ces deux classes, probablement en raison de leur faible représentation ou de similitudes avec d'autres catégories.

4 - Conclusion

Le modèle utilisant TF-IDF est préférable, car il exploite mieux les poids des termes pour différencier les classes, particulièrement dans un contexte où les classes sont déséquilibrées ou partagent des similarités textuelles.