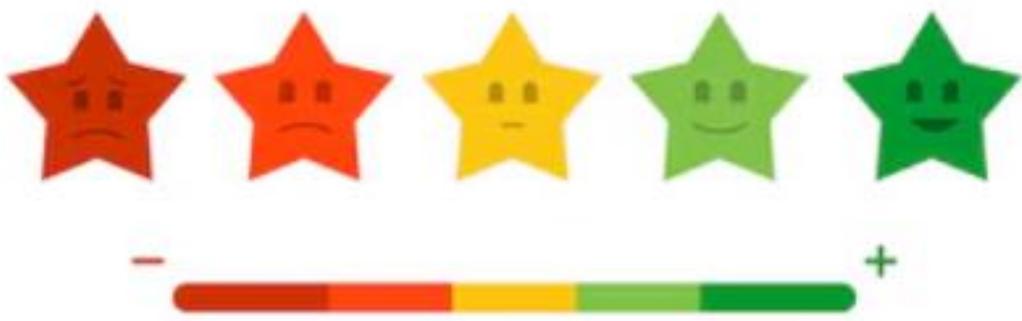




SUPPLY CHAIN SATISFACTION DES CLIENTS



Projet Datascientist 2024-2025

Remerciements

Nous tenons à remercier Eliott pour son soutien tout au long de ce projet.

Son expertise, ses conseils et sa disponibilité ont grandement contribué à la réussite de nos efforts.

Merci Eliott !

Table des matières

1.	Contexte.....	5
1.1.	Contexte d'insertion du projet dans votre métier	5
1.2.	Du point de vue technique	5
1.3.	Du point de vue économique	5
1.4.	Du point de vue scientifique.....	5
2.	Objectif	6
3.	Problématique	6
4.	Création de la Base de données	6
5.	Exploration du jeu de données	8
5.1.	Nom de l'entreprise.....	8
5.2.	Page.....	8
5.3.	Note globale :	8
5.4.	Nombre d'avis total	8
5.5.	Nom du client	8
5.6.	Pays	8
5.7.	Date de publication.....	9
5.8.	Nombre d'avis du client	9
5.9.	La note client.....	10
5.10.	Titre commentaire :.....	10
5.11.	Commentaires	11
5.12.	Score polarité.....	12
5.13.	Note client- Entreprises	13
6.	Pre-processing (Text Mining).....	14
6.1.	Suppression des doublons	14
6.2.	Traitement des valeurs manquantes	14
6.3.	Traitement des formats.....	14
6.4.	Pré-traitement	14
6.5.	Création du jeu de données pour l'entraînement, le test et la validation	15
7.	Feature engineering	16
7.1.	Suppression de colonnes	16
7.2.	Analyse de la relation entre la variable cible et les variables explicatives initiales	17
7.3.	Création de nouvelles variables	20
7.4.	Vérification de la relation entre la variable cible et les nouvelles variables créées	21

8.	Visualisations et Statistiques	25
8.1.	Analyse descriptive	25
8.2.	WordCloud	28
9.	Modélisation.....	31
9.1.	Préparation des données pour l'intégration au modèle.....	31
9.2.	Évaluation des modèles de classification.....	32
9.3.	Synthèse des résultats des modèles.....	40
10.	Ajustement du modèle	41
10.1.	Intégration de nouvelles caractéristiques NLP ("Natural Language Processing")	41
10.2.	Évaluation des modèles avec l'intégration des nouvelles caractéristiques	41
10.2.	Évaluation du modèle exclusivement sur les métadonnées	44
10.3.	Classification de la variable cible en trois catégories.....	46
10.4.	Classification de la variable cible en variable binaire.....	47
11.	Modélisation avec deep learning.....	51
12.	Conclusion	54

1.Contexte

1.1. Contexte d'insertion du projet dans votre métier

Dans un environnement commercial de plus en plus compétitif, la satisfaction client est devenue un enjeu majeur pour les entreprises. Les commentaires et avis des utilisateurs constituent une source d'information précieuse pour comprendre les attentes, les préférences et les frustrations des clients. Dans le domaine de la data science, la prédiction de la satisfaction client est une composante essentielle pour les entreprises, quel que soit leur secteur d'activité. L'analyse des retours clients permet de mieux comprendre les attentes des consommateurs, d'optimiser les services existants et d'améliorer les futures expériences utilisateur. En intégrant des feedbacks, les entreprises peuvent adapter leur offre et renforcer leur position sur le marché.

1.2. Du point de vue technique

Sur le plan technique, nous allons entreprendre un processus structuré pour prédire la satisfaction client. Nous commencerons par extraire des avis sur un site web en utilisant une technique de web scraping, suivie du nettoyage et de l'analyse des données textuelles ainsi que d'autres variables pertinentes. Nous enrichirons ensuite notre ensemble de données par la création de nouvelles variables qui pourraient influencer la satisfaction. Nous utiliserons des techniques de traitement de Text mining et d'analyse des sentiments pour extraire des insights significatifs, suivies de méthodes de régression et de machine learning pour développer des modèles prédictifs robustes.

1.3. Du point de vue économique

Économiquement, ce projet présente un fort potentiel de rentabilité.

En réduisant le temps de traitement et d'analyse des commentaires, les entreprises peuvent rapidement réagir aux préoccupations des clients et apporter des améliorations ciblées.

De plus, la capacité à anticiper les tendances de satisfaction permet d'ajuster les stratégies marketing permettant de réduire les coûts liés au service client (gestions des plaintes, retours produits ...), de minimiser le taux d'attrition, et de conserver une bonne image.

L'amélioration continue de la satisfaction client conduit à une fidélisation des clients, augmentant ainsi les revenus à long terme de l'entreprise.

1.4. Du point de vue scientifique

Sur le plan scientifique, notre projet s'inscrit dans une démarche d'innovation. Nous exploiterons des méthodologies de traitement du langage naturel et d'analyse des sentiments, pour développer un modèle capable de saisir les émotions des clients exprimées à travers les commentaires. Ce projet constitue une opportunité d'enrichir notre compréhension de la relation entre les retours des clients et leur satisfaction.

2. Objectif

L'objectif principal de ce projet est de développer un modèle de machine learning capable d'analyser et de prédire la satisfaction client à partir des commentaires des utilisateurs. En analysant les données textuelles des avis ainsi que d'autres indicateurs pertinents, notre objectif est de déceler des tendances positives ou négatives et de prédire le niveau de satisfaction des clients, à travers une note, qui sera ensuite exprimée sous forme d'étoiles.

Pour atteindre cet objectif, nous allons :

- Collecter et prétraiter les données concernant les avis sur le site Truspilot
- Analyser le contenu des commentaires
- Intégrer des indicateurs clés et des métriques adaptées telles que le F1-Score et la matrice de confusion
- Développer un modèle prédictif de classification supervisée
- Valider et optimiser le modèle

En réalisant ces étapes, nous pourrons répondre à la problématique.

3. Problématique

Comment améliorer la satisfaction client en analysant rapidement les commentaires des utilisateurs et en anticipant les tendances de satisfaction à l'aide d'indicateurs pertinents ?

4. Crédit de la Base de données

Nous avons donc décidé de créer notre propre base de données car nous n'avons pas trouvé de jeu de données existant correspondant à nos critères.

Nous avons recueilli des informations sur le site Trustpilot, une plateforme dédiée aux avis sur les entreprises. Ainsi, nous avons constitué un ensemble de données comprenant 33 563 avis concernant trois sociétés de la catégorie ordinateur et téléphone qui sont les suivantes :



- La constitution de notre jeu de données a été effectuée à l'aide de la méthode de web scraping

- L'accès aux données sur le site de Trustpilot est libre, sous réserve d'effectuer des requêtes avec un maximum de 200 pages.
- Les données récupérées :

1	Nom entreprise
2	Page
3	Note globale
4	Nombre total avis
5	Nom du client
6	Pays
7	Date de publication
8	Nombre d'avis du client
9	Note client
10	Titre commentaire
11	Commentaire

- Notre DataFrame : un aperçu des premières lignes

	Nom entreprise	Page	Note globale	Nombre total avis	Nom du client	Pays	Date de publication	Nombre d'avis du client	Note client	Titre commentaire	Commentaire
0	Materiel.net	https://fr.trustpilot.com/review/www.materiel...	4.7	29384	Charles	FR	2024-12-11	6	5	Très bon support technique	Bonjour, j'ai acheté ma première configurz...
1	Materiel.net	https://fr.trustpilot.com/review/www.materiel...	4.7	29384	Jimbo	FR	2024-12-02	2	5	Site clair et infos constructives	Je partais à la recherche de mon premier ecran...
2	Materiel.net	https://fr.trustpilot.com/review/www.materiel...	4.7	29384	paul vella	FR	2024-11-28	2	5	J'ai récemment eu un problème avec ma carte gr...	J'ai récemment eu un problème avec ma carte gr...
3	Materiel.net	https://fr.trustpilot.com/review/www.materiel...	4.7	29384	Tiphaine L.	FR	2024-11-04	2	5	Ras comme toujours	Je passe par materiel.net car historiquement j...
4	Materiel.net	https://fr.trustpilot.com/review/www.materiel...	4.7	29384	barreau	FR	2024-12-03	1	5	cela fait des années que j'achète sur...	cela fait des années que j'achète sur votre si...

5.Exploration du jeu de données

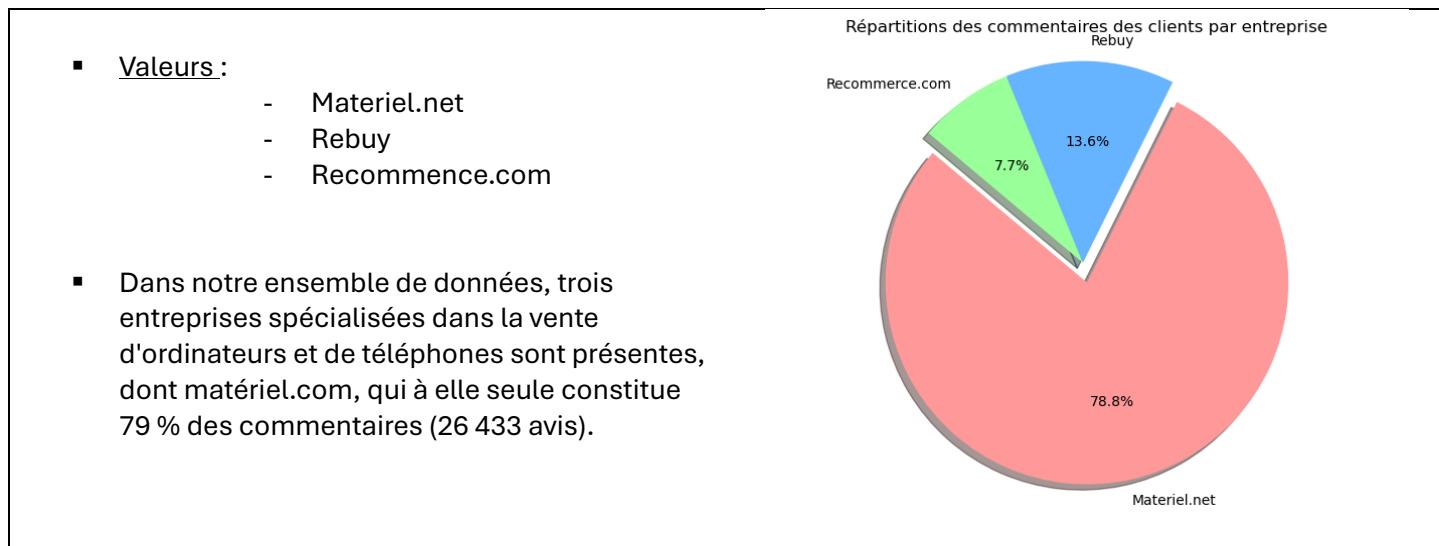
Notre jeu de données est constitué de 33 563 avis et 11 variables ce qui représente une quantité d'information intéressante pour réaliser notre projet, qui a pour objectif de prédire le nombre d'étoiles attribuées par les clients en fonction du contenu de leurs commentaires.

Base	
Nombre d'avis	33 563
Nombre de doublons	40
Nombre de valeurs manquantes	3 206
Nombre de colonnes	11

Pour atteindre nos objectifs, les variables les plus significatives seront la note attribuée par le client, qui représente notre variable cible, ainsi que le nombre d'avis publiés par ce même client, le titre du commentaire et le contenu du commentaire.

Nous allons explorer ces variables ci-dessous.

5.1. Nom de l'entreprise



5.2. **Page** : URL du commentaire

5.3. **Note globale** : Note globale par entreprise

5.4. **Nombre d'avis total** : Nombre total d'avis par entreprise

5.5. **Nom du client**

5.6. **Pays** :

Pays où le commentaire a été rédigé.

Les publications de commentaires proviennent majoritairement de la France :

30 57 avis sont issus de la France, soit 92 % du jeu de données.

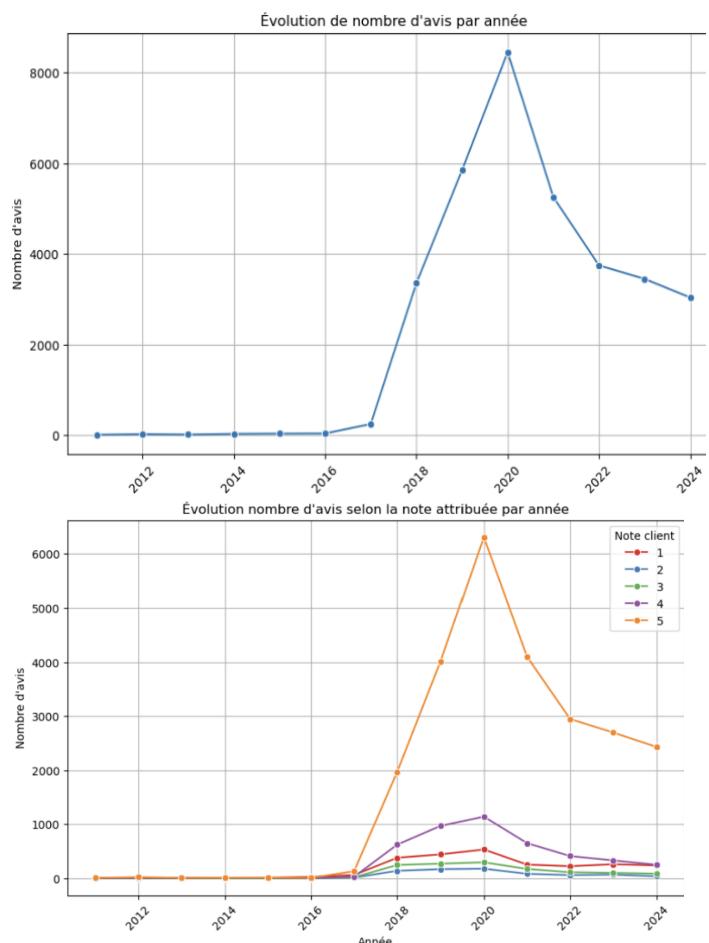
5.7. Date de publication

- Valeurs :
 - Date de 2012 à 2020
- Les données sont disponibles depuis 2012 jusqu'à aujourd'hui, avec un pic notable d'avis en 2020 : plus de 8 000 avis ont été publiés.

Ce pic peut s'expliquer par l'augmentation des achats en ligne en raison des mesures de restrictions mises en place au début de la période COVID-19.

Sur les plus de 8 000 avis publiés en 2020, plus de 6 000 ont reçu la note maximale de 5 étoiles.

- Une concentration d'avis en 2020
- L'évolution des notes suit la même tendance au fil des années



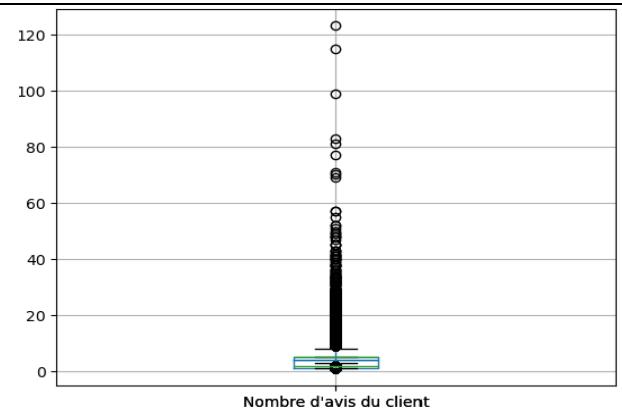
5.8. Nombre d'avis du client

- Valeurs : nombre de 1 à 123
- Il s'agit du nombre d'avis laissés par les clients sur le site.
En moyenne, un client rédige trois commentaires, avec un maximum de 123 avis.
La répartition des avis inclut des valeurs extrêmes qui semblent plausibles et cohérentes, mais à traiter.

Statistiques sur le nombre d'avis déposés

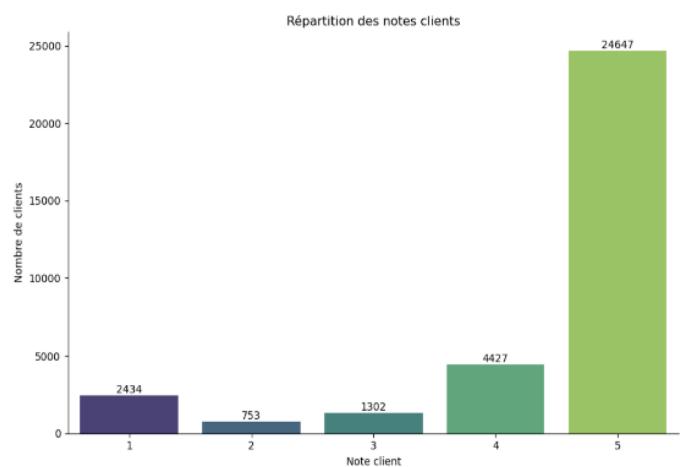
count	33563.000000
mean	3.166344
std	4.000113
min	1.000000
25%	1.000000
50%	2.000000
75%	4.000000
max	123.000000
Name:	Nombre_avis_client, dtype: float64

- Cette variable présente une large amplitude, variant de 1 à 123 avis.



5.9. La note client

- Valeurs :
 - Les notes vont de 1 à 5,
 - où « 5 » indique une satisfaction élevée.
- La distribution des notes révèle que 73 % des évaluations sont classées 5 étoiles, tandis que seulement 7 % ont obtenu la note la plus basse de 1 étoile. Ces résultats témoignent d'un sentiment globalement positif de la part des clients.
- Notre jeu de données présente un déséquilibre, avec une dominance d'avis positifs, ce qui pourrait affecter les résultats du modèle.



5.10. Titre commentaire :

La longueur maximale d'un titre est : 200 caractères.

5.11. Commentaires

▪ Valeurs :

- Nombre de mots au total : 1 040 221
- Nombre de mots distinct : 44 942

▪ Taille du commentaire :

Cette répartition montre une grande variabilité dans la longueur des commentaires, allant d'un mot à près de 1 000 mots par commentaire.

- Cette variabilité peut avoir un impact significatif sur la qualité et la pertinence des informations extraites des commentaires.**

Statistiques sur le nombre de mots par commentaire

```
count      33563.000000
mean       30.993088
std        45.552534
min        1.000000
25%        9.000000
50%        17.000000
75%        35.000000
max        980.000000
Name: nbre_mots, dtype: float64
```

▪ Les point d'exclamation par commentaire :

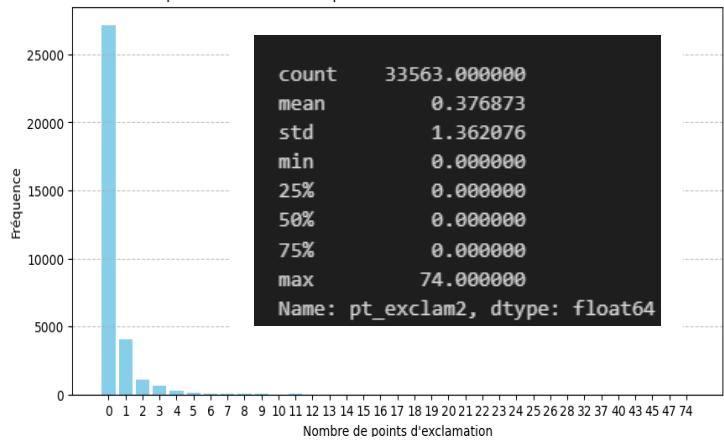
▪ Valeurs : Nombre de points d'exclamation

Nous avons identifié le nombre de points d'exclamation dans les commentaires.

20 % des commentaires contiennent des points d'exclamation, variant de 1 à 74 points.

Toutefois, la majorité d'entre eux se situe principalement entre 1 et 3 points d'exclamation par commentaire.

Répartition du nombre de points d'exclamation dans les commentaires



Point d'exclamation dans les commentaires		
Non	27 128	80,8%
Oui	6 435	19,2%
Total	33 563	100%

- Les émoticônes dans les commentaires :

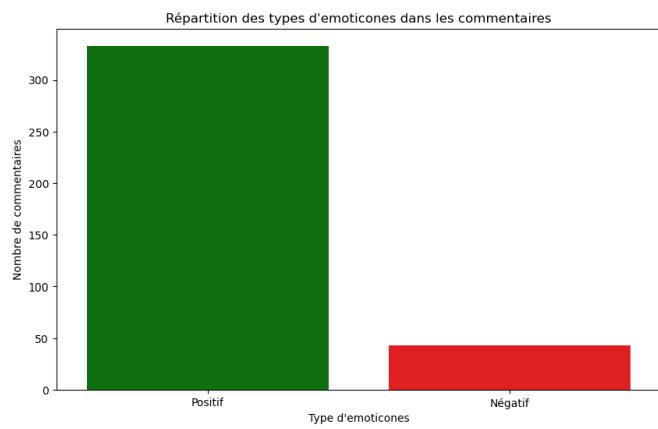
- Valeurs :

- Positif
- Négatif

À travers les commentaires, nous avons observé que 2 % d'entre eux contenaient des émoticônes.

Ces émoticônes ont été regroupées en deux catégories distinctes : positives ou négatives, afin d'évaluer le sentiment dégagé par les commentaires.

Il est intéressant de noter que l'utilisation des émoticônes est majoritairement liée à des commentaires positifs.



5.12. Score polarité

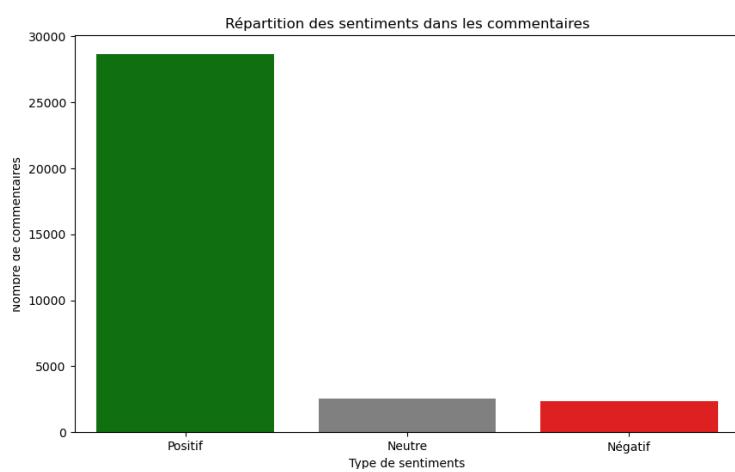
- Valeurs :

- Score entre -1 et 1

Nous appliquons la fonction TxtBlob-fr pour évaluer le sentiment exprimé dans le commentaire. Cette fonction détermine un score qui se décline :

- 1 indique un sentiment très négatif
- 0 donne un sentiment neutre
- 1 reflète un sentiment très positif.

Pour faciliter la visualisation de la distribution de ces scores, nous les regrouperons selon ces catégories.

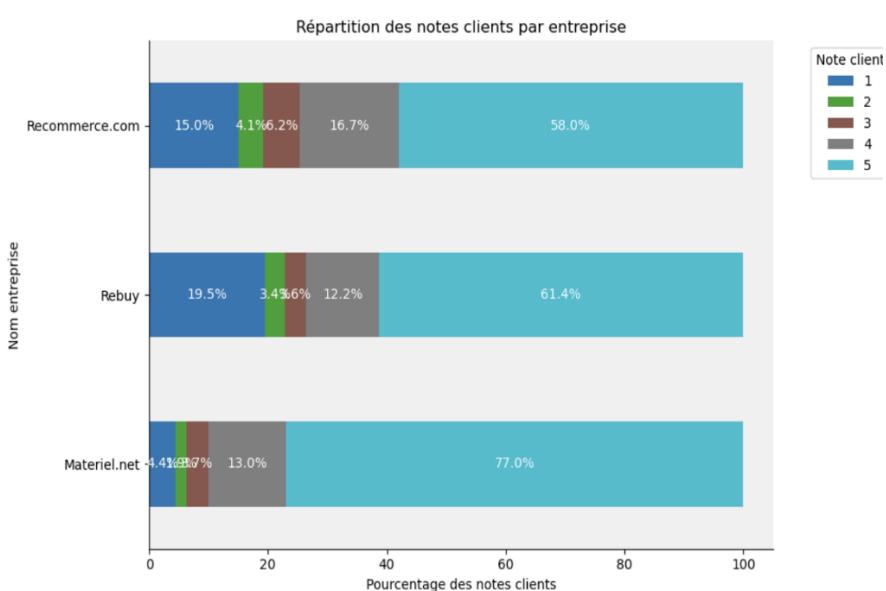


5.13. Note client- Entreprises

Le site Materiel.net se démarque par la proportion la plus élevée de commentaires clients à 5 étoiles, avec 77 %, suivi de Rebuy à 61,4 % et Recommerce.com à 58 %.

En ce qui concerne les évaluations à 1 étoile, Rebuy affiche la plus grande part avec 19,5 %, suivie par Recommerce.com à 15 % et Materiel.net à 4,41 %.

Malgré de légères variations dans les évaluations, on constate une répartition des notes relativement similaire entre les trois entreprises.



- Par conséquent, dans la suite du projet, nous aborderons notre analyse sans faire de distinction entre les entreprises.

6. Pre-processing (Text Mining)

Notre base de données était relativement propre, car nous l'avons construite nous-mêmes grâce au Web scraping. Cela nous a permis de sélectionner uniquement les informations pertinentes pour notre projet.

Après l'extraction, nous avons effectué un nettoyage pour éliminer les doublons, traiter les valeurs manquantes et harmoniser les formats.

6.1. Suppression des doublons

Nous avons ainsi supprimé 40 doublons.

6.2. Traitement des valeurs manquantes

Ensuite, nous avons traité les 3 206 commentaires manquants. Pour éviter de perdre des informations, nous avons choisi de remplacer ces commentaires absents par le titre commentaire associé, en supposant que les clients n'avaient pas pris le temps de rédiger un avis détaillé et que le titre reflétait globalement leur satisfaction.

6.3. Traitement des formats

Enfin, nous avons converti les dates en format 'datetime', car elles étaient initialement au format 'object' et contenaient des accents non reconnus.

6.4. Pré-traitement

6.4.1. Normalisation :

Les commentaires et les titres de commentaires ont été convertis en minuscules.

En revanche, à ce stade, nous avons décidé de conserver les caractères spéciaux, car nous allons les analyser. Ils pourraient en effet nous apporter des éléments de compréhension sur ce qui détermine si un commentaire est positif ou négatif.

6.4.2. Tokenisation :

Nous avons découpé les commentaires et leurs titres en mots et en phrases afin de faciliter leur analyse et leur traitement.

6.4.3. Stop words : Suppression des mots vides

L'objectif est de retirer les termes courants qui n'apportent pas d'informations significatives à l'analyse.

En fonction de nos nuages de mots, nous avons inclus dans la liste des mots à supprimer des termes tels que "cela", "fait", "dire", "bonjour", "faire", "avoir", "peut", "donc", "mais", etc.

6.4.4. Filtrage :

Nous avons conservé les mots ayant plus de 3 caractères.

6.4.5. Lemmatisation :

Ensuite, nous avons appliqué le processus de lemmatisation pour simplifier et regrouper les mots en deux catégories : les verbes et les noms ayant un sens similaire.

6.5. Création du jeu de données pour l'entraînement, le test et la validation

Dans le cadre de notre projet, nous avons choisi d'utiliser la méthode de hold-out pour partitionner notre ensemble de données en trois segments distincts :

- 80 % pour l'entraînement,
- 10 % pour les tests
- 10 % pour la validation des hyperparamètres.

Cette approche nous permet de garantir que le modèle est bien entraîné et évalué de manière rigoureuse.

Habituellement, nous sauvegardons les données brutes dans trois fichiers distincts respectivement nommés :

- train_projet_truspilot2_etape2.csv
- test_projet_truspilot2_etape2.csv
- validation_projet_truspilot2_etape2.csv

Cependant, dans le contexte de notre projet, nous avons décidé de réaliser cette séparation ultérieurement. Nous estimons que notre ensemble de données ne présentera pas de fuite d'informations, ce qui nous permet de retarder cette étape.

En effet, il est courant de séparer les données avant d'appliquer un rééquilibrage de la variable cible, afin d'éviter toute contamination entre les ensembles d'entraînement et de test.

7. Feature engineering

Pour améliorer notre modèle, certaines colonnes ont été supprimées et des caractéristiques (features) ont été créées à partir des données existantes.

7.1. Suppression de colonnes

Nous supprimons les données ci-dessous car elles étaient utiles dans un premier temps pour valider la récupération des informations via la méthode web scraping et pour comprendre notre jeu de données. Cependant, elles ne sont pas pertinentes pour notre future modélisation.

- **Nom de l'entreprise** : comme mentionné précédemment, il y a peu de variations dans les évaluations entre les trois sociétés. Par conséquent, « **Nom de l'entreprise** » ne constitue pas une variable pertinente pour établir la note de prédiction.
- **Page** : URL du commentaire utile seulement pour la vérification des données scrappées.
- **Note globale** : Étant donné que la variable « **Nom de l'entreprise** » n'est pas incluse dans le dispositif de prévision, la note globale de l'entreprise ne doit pas être conservée.
- **Nom du client** : Cette donnée est mal renseignée. De plus, il est impératif d'anonymiser notre jeu de données afin de respecter les exigences du RGPD. Il aurait été pertinent d'avoir un numéro d'identification unique pour permettre de regrouper les avis émis par un même client. Cela aurait amélioré notre analyse tout en garantissant la protection des données personnelles.
- **Pays** : 92 % des commentaires proviennent de la France. Dans ce contexte, la variable "pays" perd de son intérêt pour les prédictions de la note de satisfaction. Si les données avaient été équilibrées entre plusieurs pays, cette variable aurait pu fournir des informations pertinentes. Telle quelle, elle pourrait introduire un biais plutôt que d'enrichir notre modèle. Par conséquent, nous la supprimons.
- **Date de publication** : Comme cela a été souligné précédemment, les notes attribuées par les clients suivent la même tendance au fil des années. Cette information ne nous aidera pas à affiner notre modèle.

7.2. Analyse de la relation entre la variable cible et les variables explicatives initiales

Nous procémons à une analyse plus approfondie de la relation entre la variable cible, qui est la note de satisfaction, et les variables explicatives initiales. Cette étape est importante pour affiner notre modèle, car elle nous permet de mieux comprendre les facteurs qui influencent la satisfaction des utilisateurs.

À partir de notre analyse,

- Nous identifierons les variables les plus significatives qui ont un impact sur la satisfaction,
- Nous éliminerons celles qui sont redondantes ou non pertinentes,
- Nous développerons de nouvelles variables dérivées des variables initiales.

7.2.1. Note client et nombre d'avis du client

On constate qu'il existe une corrélation significative entre la note client et le nombre d'avis laissés par les clients sur le site grâce au test Anova.

Relation entre la note et le nombre d'avis client (par Test d'ANOVA)

```
Test ANOVA

H0 : Il n'y a pas d'effet significatif de la variable nombre d'avis sur la note
H1 : Il y a un effet significatif de la variable nombre d'avis sur la note

La statistique du test est : 36.99643972907156
La p-valeur du test est : 1.1962219964896114e-09

Conclusion: le nombre d'avis a un effet significatif sur la note du client
```

Comme nous l'avons constaté précédemment, la variable « nombre d'avis » présente une large amplitude, variant de 1 à 123 avis.

Pour faciliter l'analyse et mieux comprendre le comportement des utilisateurs qui laissent des avis, nous simplifierons cette variable en classe.

7.2.2. Note client et le nombre de mot dans les commentaires

On constate que la note attribuée par les clients et la taille des commentaires sont étroitement liés selon le test d'Anova.

Relation entre la taille du commentaire et la note du client (par Test d'ANOVA)

	df	sum_sq	mean_sq	F	PR(>F)
Note_client	1.0	1.435318e+07	1.435318e+07	8712.513727	0.0
Residual	33561.0	5.528909e+07	1.647421e+03	NaN	NaN

Test ANOVA

H0 : Il n'y a pas d'effet significatif de la variable nombre de mot sur la note
H1 : Il y a un effet significatif de la variable nombre de mot sur la note

La statistique du test est : 8712.513726854979

La p-valeur du test est : 0.0

Conclusion: le nombre de mot dans le commentaire a un effet significatif sur la note du client

Comme cela a été souligné précédemment, la variable « taille du commentaire » présente une grande variabilité. Certains utilisateurs peuvent donner des avis très courts, tandis que d'autres peuvent rédiger des commentaires beaucoup plus longs. Cette variabilité peut rendre difficile l'établissement d'une relation claire entre la taille du commentaire et la note de satisfaction.

C'est pourquoi nous regrouperons le nombre de mots dans les commentaires dans une nouvelle variable. Elle nous permettra d'améliorer notre compréhension du contenu, d'enrichir notre modèle prédictif et de découvrir des insights précieux sur les comportements des clients.

7.2.3. Note client et le nombre phrase dans les commentaires

On constate que la note attribuée par les clients et le nombre de phrase dans les commentaires sont liés d'après le test d'Anova.

Test ANOVA

H0 : Il n'y a pas d'effet significatif de la variable nombre de phrase sur la note
H1 : Il y a un effet significatif de la variable nombre de phrase sur la note

La statistique du test est : 3703.2315305538204

La p-valeur du test est : 0.0

Conclusion: le nombre de phrase dans le commentaire a un effet significatif sur la note du client

7.2.4. Note client et le nombre de points d'exclamation dans les commentaires

On constate que la note attribuée par les clients et le nombre de points d'exclamation dans les commentaires sont liés d'après le test d'Anova.

Relation entre le nombre de points d'exclamation et la note du client (par Test d'ANOVA)

	df	sum_sq	mean_sq	F	PR(>F)
Note_client	1.0	2648.372869	2648.372869	1490.870265	3.720166e-319
Residual	33561.0	59617.556220	1.776394	Nan	Nan

Test ANOVA

H0 : Il n'y a pas d'effet significatif de la variable nombre de point d'exclamation sur la note
H1 : Il y a un effet significatif de la variable nombre de point d'exclamation sur la note

La statistique du test est : 1490.870265224722
La p-valeur du test est : 3.72017e-319

Conclusion: le nombre de point d'exclamation dans le commentaire a un effet significatif sur la note du client

Nous utiliserons le nombre de points d'exclamation comme caractéristique dans notre modèle de prédiction de notes, car il existe une corrélation significative entre la note attribuée par le client et le nombre de points d'exclamation mais nous devons réduire son amplitude.

7.2.5. Note client et le score de polarité

On constate que la note attribuée par les clients et le score de polarité dans les commentaires sont liés selon le test d'Anova.

Relation entre le score de polarité et la note du client (par Test d'ANOVA)

Test ANOVA
H0 : Il n'y a pas d'effet significatif de la variable score de polarité sur la note
H1 : Il y a un effet significatif de la variable score de polarité sur la note
La statistique du test est : 5928.133650735545
La p-valeur du test est : 0.0
Conclusion: le score de polarité a un effet significatif sur la note du client

Comme nous l'avons observé précédemment, le score de polarité est plus clair lorsqu'il est associé à des regroupements de tonalités de sentiments, ce qui facilite également son interprétation. Il serait donc pertinent de l'intégrer à notre modèle en le classifiant.

7.3. Création de nouvelles variables

7.3.1. Variable : class_nbavis

Nous avons décidé de classifier la variable représentant le nombre d'avis déposés par les clients, car elle est étroitement liée à notre variable cible, la note. Nous allons créer trois classes basées sur les quartiles.

Class_nbavis :	Classification du nombre d'avis selon les quartiles.
1	1
2-3	2-3
4-123	4-123

7.3.2. Variable : class_longueur_mot

Nous avons créé 4 classes basées sur la fréquence du nombre de mots

Class_longueur_mot :	Classification du nombre de mots dans les commentaires selon les quartiles
« très court »	"1 à 9 mots
« court »	"10 à 17 mots
« moyen »	"18 à 35 mots
« long »	"35 à 980 mots

7.3.3. Variable : nbre_phrases

Nombre de phrase par commentaire

7.3.4. Variable : class_pt_exclam

Nous avons créé 5 classes basées sur la répartition du nombre de points d'exclamation

Class_pt_exclam:	Classification des occurrences de points d'exclamation dans les commentaires par quartiles.
« E0 »	0 point d'exclamation
« E1 »	1 point d'exclamation
« E2 »	2 points d'exclamation
« E3 »	3 points d'exclamation
« E4 »	4 et plus points d'exclamation

7.3.5. Variable : émoticônes et nombre d'émoticônes

Nous avons analysé les émoticônes présents dans les commentaires et les avons classés selon leur nature, qu'ils soient positifs ou négatifs. Dans le cas où un même commentaire contient à la fois des émoticônes positifs et négatifs, nous attribuons la tonalité positive si les émoticônes positifs sont plus nombreux, et inversement.

Emoticônes	
Positif	"{:sunglasses:}", "{:ok:}", "{:smile:}", "{:smiling:}", "{:grinning:}", "{:laughing:}", "{:hourglass:}", "{:alarm_clock:}", "{:blush:}", "{:cool:}", "{:sparkles:}", "{:heart:}", "{:handshake:}", "{:star:}", "{:wink:}", "{:clapping:}", "{:grinning:}", "{:sweat_smile:}.....
Négatif	"{:angry:}", "{:fist:}", "{:crying:}", "{:sob:}", "{:neutral:}", "{:disappointed:}", "{:confused:}", "{:sick:}", "{:sweat:}", "{:angry:}", "{:broken_heart:}", "{:frowny:}", "{:sick:}", "{:sob:}", "{:neutral:}", "{:disappointed:}", "{:sick:}", "{:no_good:}", "{:sweat_smile:}", "{:sick:}", "{:sob:}", "⚠", "{:no_good:}", "{:no_good:}", "🚫".....

7.3.6. Variable : class_sentiment

Pour une meilleure interprétation du score de polarité, nous le classifions de la manière suivante :

class_sentiment	Score
Négatif	Score <0
Neutre	Score=0
Positif	Score>0

7.4. Vérification de la relation entre la variable cible et les nouvelles variables créées

Nous allons examiner la relation entre les nouvelles variables créées, qui sont dérivées de celles que nous avons déjà analysées, afin de déterminer si ces variables sont pertinentes pour prédire la note et si elles doivent être intégrées dans un modèle de prédiction.

7.4.1. Note client et classe nombre d'avis

Selon les analyses des tableaux croisés et le test Khi2 d'indépendance, la note attribuée par les clients et la classe du nombre d'avis sont liés.

Il semble qu'un grand nombre d'avis laissés par les clients reflète un sentiment de mécontentement.

Note_client	1	2	3	4	5	
class_nbavis						
1	0.080362	0.026356	0.042074	0.138681	0.712526	
2-3	0.062265	0.017522	0.034418	0.120932	0.764862	
4-123	0.056957	0.014438	0.032366	0.120578	0.775662	
Total	0.072520	0.022435	0.038793	0.131901	0.734350	

Note_client	1	2	3	4	5	Total
class_nbavis						
1	0.688989	0.730412	0.674347	0.653716	0.603278	0.621756
2-3	0.163517	0.148738	0.168971	0.174610	0.198361	0.190448
4-123	0.147494	0.120850	0.156682	0.171674	0.198361	0.187796

Relation entre la classe nombre d'avis du client et la note du client (par Test d'Khi 2)

	df	sum_sq	mean_sq	F	PR(>F)
Note_client	1.0	1.435318e+07	1.435318e+07	8712.513727	0.0
Residual	33561.0	5.528909e+07	1.647421e+03	NaN	NaN

Test ANOVA

H0 : Il n'y a pas d'effet significatif de la variable nombre de mot sur la note
H1 : Il y a un effet significatif de la variable nombre de mot sur la note

La statistique du test est : 8712.513726854979
La p-valeur du test est : 0.0

Conclusion: le nombre de mot dans le commentaire a un effet significatif sur la note du client

7.4.2. « Note client » et classe « longueur du commentaire »

Selon les analyses de tableaux croisés et le test Khi2 d'indépendance, la note attribuée par les clients et la taille des commentaires sont liés.

Il semble que les commentaires longs soient associés à des avis négatifs

Note_client	1	2	3	4	5	Total
class_longueur_mot						
Très court	0.053821	0.046481	0.057604	0.213463	0.325719	0.274528
Court	0.071898	0.081009	0.093702	0.184775	0.279060	0.239967
moyen	0.156532	0.221780	0.259601	0.272193	0.239177	0.237941
long	0.717749	0.650730	0.589094	0.329569	0.156043	0.247564

Relation entre la classe longueur du commentaire et la note du client (par Test dKhi 2)

Test du khi2 d'indépendance

H_0 : La note du client est indépendante de la longueur du commentaire
 H_1 : La note du client n'est pas indépendante de la longueur du commentaire

La statistique du test est : 6116.871301280829

La p-valeur du test est : 0.0

Nous rejetons l'hypothèse nulle H_0 .

La note du client dépend de la longueur du commentaire.

7.4.3. « Note client » et classe « nombre de points d'exclamation »

Selon les analyses des tableaux croisés et le test Khi2 d'indépendance, la note attribuée par les clients et le nombre de points d'exclamation dans les commentaires sont liés. Il semble que les commentaires comportant plus de trois points d'exclamation soient liés à des avis négatifs, tandis que ceux avec un ou deux points d'exclamation tendent à être associés à des avis positifs.

Note_client	1	2	3	4	5
class_pt_exclam					
E0	0.055662	0.021196	0.040807	0.143210	0.739126
E1	0.069916	0.020433	0.028557	0.091334	0.789759
E2	0.143123	0.028810	0.032528	0.078996	0.716543
E3	0.235387	0.039494	0.036335	0.085308	0.603476
E4	0.505295	0.059002	0.031770	0.048411	0.355522
Total	0.072437	0.022437	0.038796	0.131913	0.734416

Note_client	1	2	3	4	5	Total
class_pt_exclam						
E0	0.621144	0.763612	0.850230	0.877569	0.813527	0.808343
E1	0.116824	0.110226	0.089094	0.083804	0.130158	0.121037
E2	0.063348	0.041169	0.026882	0.019200	0.031282	0.032062
E3	0.061292	0.033201	0.017665	0.012198	0.015499	0.018862
E4	0.137392	0.051793	0.016129	0.007228	0.009535	0.019696

Relation entre la classe longueur du commentaire et la note du client (par Test dKhi 2)

Test du khi2 d'indépendance

H_0 : La note du client est indépendante du nombre de point d'exclamation

H_1 : La note du client n'est pas indépendante du nombre de points d'exclamation

La statistique du test est : 2485.5459349725406

La p-valeur du test est : 0.0

Nous rejetons l'hypothèse nulle H_0 .

La note du client dépend du nombre de points d'exclamation.

7.4.4. « Note client » et classe « des émoticônes »

À la suite du test de khi2 et le tableau croisé ci-dessous, qui a pour but de vérifier s'il existe une corrélation entre la note client et la présence d'émoticônes dans le commentaire, nous avons une p-value de 0,000153 (donc <0.05), nous rejetons donc H0 et notons une corrélation entre la note client et la présence d'émoticônes dans les commentaires. Nous notons que les commentaires ayant la présence d'émoticônes sont les commentaires qui ont une tendance positive.

Présence Émoticônes \ Classe Note	Non	Oui
Faible	3142	47
Moyenne	5689	49
Elevée	24284	392

Test du khi2 d'indépendance

H_0 : La note du client est indépendante de la présence d'émoticônes dans le commentaire

H_1 : La note du client n'est pas indépendante de la présence d'émoticônes dans le commentaire

La statistique du test est : 17.566940524975568

La p-valeur du test est : 0.00015324535976142298

Nous rejetons l'hypothèse nulle H_0 .

La note du client dépend de la présence d'émoticônes dans le commentaire.

7.4.5. « Note client » et classe « sentiment »

Selon le test de Khi 2, nous pouvons affirmer que la note attribuée par les clients et la classe sentiment sont liés.

Relation entre la classe sentiment et la note du client (par Test dKhi 2)

Test du khi2 d'indépendance

H_0 : La note du client est indépendante de la classe sentiment

H_1 : La note du client n'est pas indépendante de la classe sentiment

La statistique du test est : 5400.900788056859

La p-valeur du test est : 0.0

Nous rejetons l'hypothèse nulle H_0 .

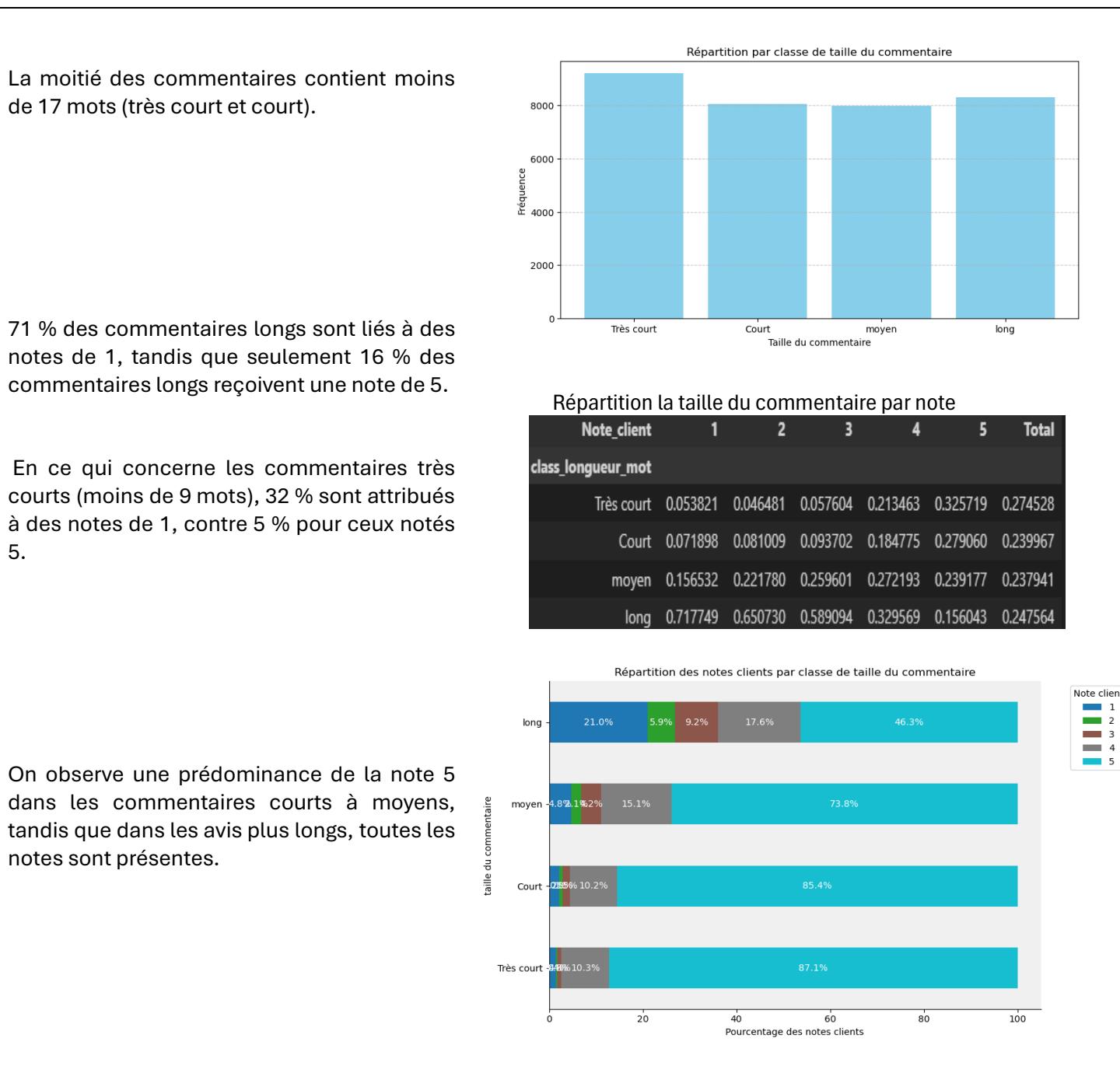
La note du client dépend de la classe sentiment.

8. Visualisations et Statistiques

8.1. Analyse descriptive

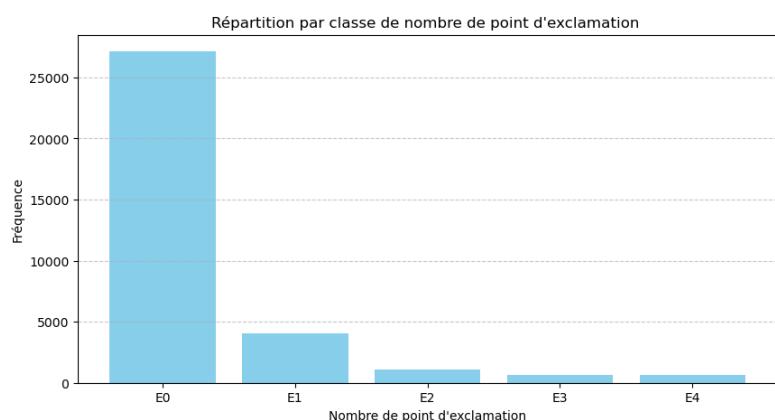
Nous disposons désormais d'une nouvelle base de données qui a été traitée et enrichie avec des variables supplémentaires. Analysons-la, car elle nous servira à élaborer notre modèle de prédiction.

8.1.2. Répartition des notes par classe de « taille du commentaire »



8.1.2. Répartition des notes par classe de « nombre de point d'exclamation »

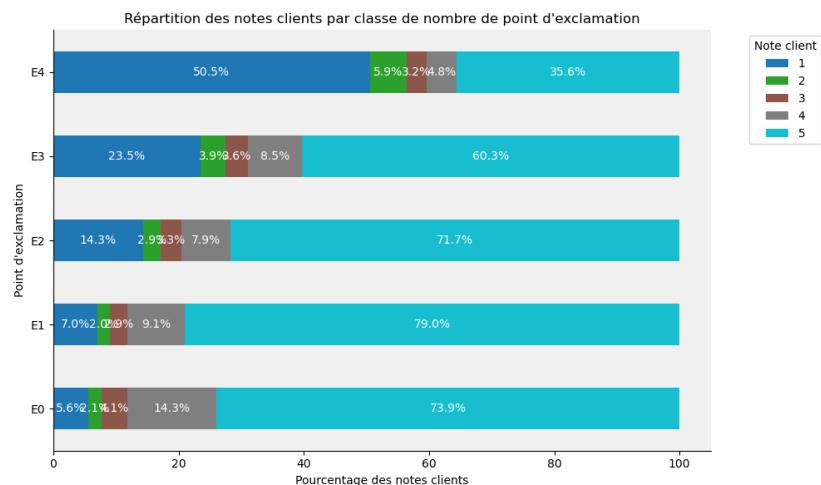
20 % des commentaires contiennent des points d'exclamation. Toutefois, la majorité d'entre eux se situe principalement entre 1 et 3 points d'exclamation par commentaire.



Répartition de la classe nombre de point d'exclamation par note

Note_client	1	2	3	4	5	Total
class_pt_exclam						
E0	0.621144	0.763612	0.850230	0.877569	0.813527	0.808343
E1	0.116824	0.110226	0.089094	0.083804	0.130158	0.121037
E2	0.063348	0.041169	0.026882	0.019200	0.031282	0.032062
E3	0.061292	0.033201	0.017665	0.012198	0.015499	0.018862
E4	0.137392	0.051793	0.016129	0.007228	0.009535	0.019696

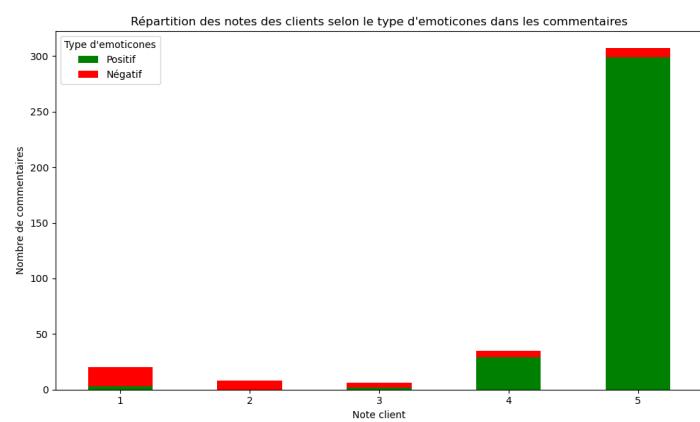
Il semble que l'utilisation de 1 à 2 points d'exclamation soit généralement liée à des avis positifs (notés 4 ou 5), tandis que les commentaires comportant plus de 3 points d'exclamation tendent à être associés à des avis négatifs.



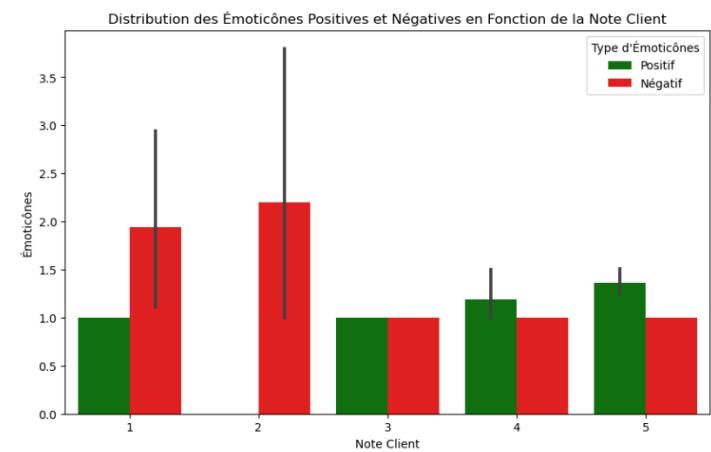
8.1.3. Répartition des notes par classe d'émoticônes

Il est intéressant de noter que l'utilisation des émoticônes est majoritairement liée à des commentaires positifs.

Nous observons sur le graphe ci-contre la répartition des émoticônes selon la note attribuée par le client. On note une prédominance des commentaires positifs, totalisant environ 350 unités, tandis que les commentaires négatifs se limitent à environ 50.

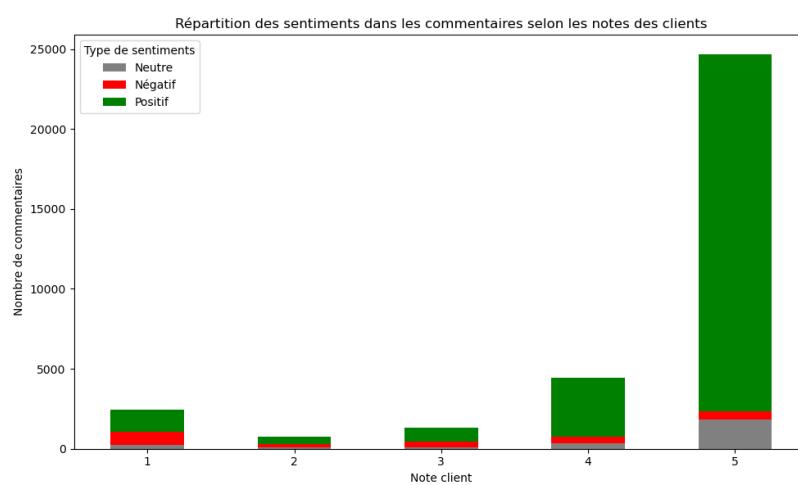


Sur le 3 -ème graphe, nous illustrons la distribution des émoticônes positifs et négatifs en fonction des notes clients (de 1 à 5). Nous remarquons que les clients insatisfaits ont tendance à mettre au moins 2 émoticônes sur leur commentaire, contrairement aux clients satisfaits qui ont tendance à mettre 1 émoticône.



8.1.4. Répartition des notes par classe sentiment

Nous observons sur le graphe ci-contre la répartition des émoticônes selon la note attribuée par des sentiments. On note une prédominance des commentaires avec un sentiment positif.



8.2. WordCloud

8.2.1. WordCloud toutes notes confondues

L'analyse des mots-clés dans les commentaires et les titres des commentaires montre une forte présence de termes positifs tels que « livraison rapide, parfait, excellent... » et des thèmes comme produit, livraison, satisfaction globale

WordCloud à partir de la colonne commentaires



Fréquence des mots

Livraison	: 10708
Rapide	: 9843
Bien	: 7743
Commande	: 7118
Matériel	: 5465
Site	: 5401
Produit	: 5232
Plus	: 4778
Colis	: 4614
Service	: 4120
Conforme	: 3875
État	: 3385
Prix	: 3377
Parfait	: 3331
Rien	: 3166
Toujours	: 3097
Recommande	: 3012
Téléphone	: 2958
Sans	: 2773
Qualité	: 2730
Merci	: 2568
Rapidement	: 2526
Prol	: 2494
Achat	: 2452
Matériel	: 2424
Temps	: 2167
Satisfait	: 2148
Jours	: 2146

WordCloud à partir de la colonne Titre des commentaires

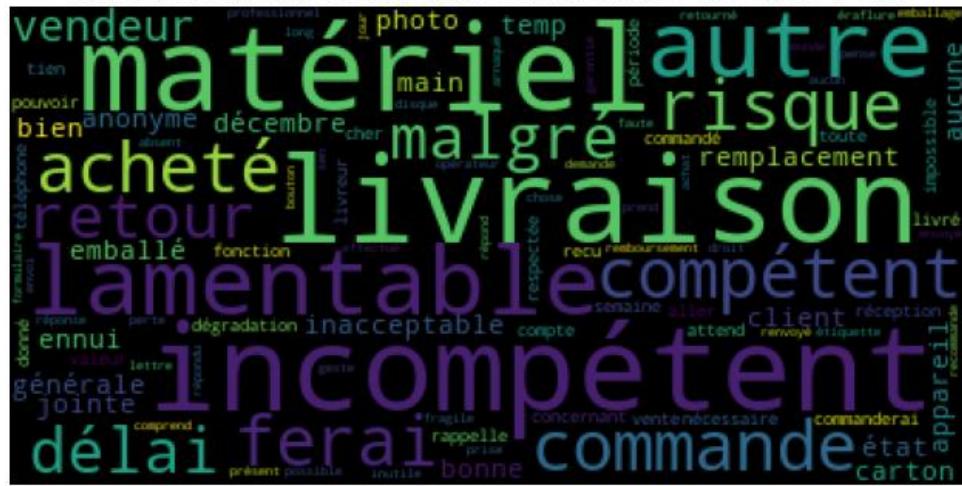


8.2.2. WordCloud avec des notes 1 et 2

L'analyse des mots-clés dans les commentaires montre une forte présence de termes négatifs tels que « incompétent, lamentable... »

Les thèmes qui ressortent sont l'absence de service client et un problème de qualité du produit.

WordCloud à partir de la colonne commentaires avec note 1 et 2



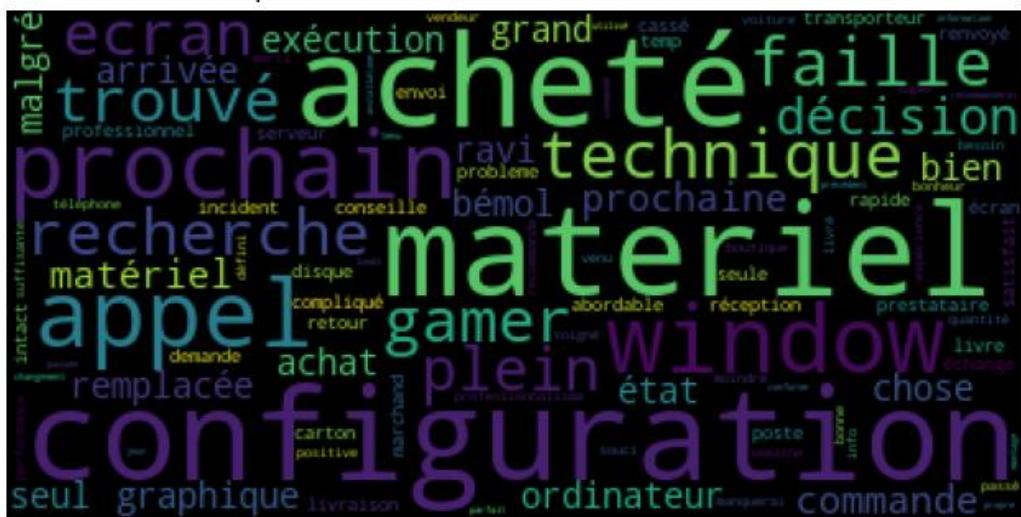
8.2.3. WordCloud avec des notes 3

WordCloud à partir de la colonne commentaires avec note 3



8.2.4. WordCloud avec des notes 4 et 5

WordCloud à partir de la colonne commentaires avec note 4 et 5



Suite à l'analyse de ces trois nuages de mots en fonction des notes données par les clients, il apparaît que les notes 1 et 2 sont associées à des termes négatifs. En revanche, les nuages correspondant aux notes 3, 4 et 5 montrent une différence peu marquée. Il est plus difficile de discerner une tonalité spécifique parmi les mots utilisés pour les notes 3, 4 et 5, ce qui pourrait poser des défis pour le modèle de prédiction des notes.

9. Modélisation

Après avoir analysé les caractéristiques de notre jeu de données pour nous assurer que les variables sont bien corrélées à la variable cible, nous allons préparer ce jeu de données pour l'intégrer dans le modèle de prédiction de notes.

Nous commencerons par diviser notre jeu de données en trois ensembles distincts : entraînement, test et validation. Puis nous poursuivrons par l'encodage numérique des variables et la normalisation des données. Nous finirons ensuite par appliquer des techniques de rééquilibrage de la variable cible sur l'ensemble d'entraînement afin d'éviter tout biais.

Enfin, nous testerons plusieurs modèles de classification, en les comparant sur la base de métriques de performance telles que l'accuracy, la F1-score et la matrice de confusion, afin de sélectionner le modèle le plus performant. Parallèlement, nous effectuerons des analyses d'erreurs pour mieux comprendre les limites du modèle et identifier des pistes d'amélioration.

9.1. Préparation des données pour l'intégration au modèle

9.1.1. Séparation du jeu de données

Nous procéderons à la séparation de notre jeu de données en 3 ensembles distincts :

- Un pour l'entraînement (80% du jeu de données)
- Un pour le test (10% du jeu de données)
- Un pour la validation (10% du jeu de données)

Comme nous l'avons mentionné précédemment, dans le cadre de notre projet, nous avons choisi d'effectuer cette séparation à ce stade, car nous pensons que notre ensemble de données ne comportera pas de fuite d'informations.

9.1.2. Encodage des variables catégorielles et textuelles

Nous commencerons par encoder les variables catégorielles afin de les transformer en un format numérique compatible pour rentrer dans l'algorithme.

Nous utilisons trois méthodes d'encodage :

- La note sera encodée à l'aide de LabelEncoder, car il s'agit d'une variable ordinaire.
- Les variables textuelles seront vectorisées à l'aide de la méthode TF-IDF.
- Les autres variables catégorielles seront encodées avec l'encodage one-hot.

9.1.3. Normalisation des données numériques

Après avoir équilibré notre variable cible de l'ensemble d'entraînement, nous normaliserons les données des 3 ensembles avec la fonction StandardScaler() pour garantir que toutes les caractéristiques sont sur la même échelle pour éviter que certaines variables ne dominent les autres en raison de leur échelle.

9.1.4. Rééquilibrage de la variable cible

Nous allons appliquer des techniques de rééquilibrage à la variable cible de l'ensemble d'entraînement, car, comme nous l'avons noté précédemment, cette variable présente un déséquilibre avec 73 % de note 5. Cela pourrait introduire des biais pouvant affecter la performance du modèle. Plusieurs méthodes sont à notre disposition, telles que le suréchantillonnage des classes minoritaires avec la technique SMOTE (Synthetic Minority Over-sampling Technique) ou le sous-échantillonnage des classes majoritaires à l'aide de la méthode Undersampling ClusterCentroid.

Dans le projet, nous allons utiliser la méthode SMOTE, qui est particulièrement adaptée aux modèles de classification multi-classes déséquilibrés.

Classes échantillon Oversampling SMOTE : {5: 19741, 4: 19741, 2: 19741, 1: 19741, 3: 19741}

9.2. Évaluation des modèles de classification

Nous allons chercher à identifier le modèle de classification le plus adapté pour prédire une note en fonction du contenu des commentaires.

Nous avons testé plusieurs modèles de classification, notamment RandomForestClassifier, GradientBoostingClassifier, Support Vector Machines, LightGBM Classifier et la régression logistique.

Dans cette approche, nous choisirons les trois modèles les plus performants pour poursuivre notre analyse : le RandomForestClassifier, le LightGBM Classifier et la régression logistique.

Nous évaluerons leur performance à l'aide de métriques telles que le F1-score, qui apporte une bonne compréhension du fonctionnement du modèle en affichant des résultats clairs tout en tenant compte du déséquilibre des classes, et l'accuracy, qui bien qu'intéressante, doit être interprétée avec prudence en raison de ce même déséquilibre. Par la suite, nous analyserons la matrice de confusion et procéderons à une analyse des erreurs.

- Rappel des variables intégrées dans le modèle :
 - **Variable cible** “Note client”

- **Variables catégorielles :** 'class_longueur_mot',"class_pt_exclam", "emoticones""class_nbavis""class_sentiment"
- **Variables texte :** commentaire + titre commentaire
- **Variables numériques :** Nombre_Emoticônes,"nbre_phrases"

9.2.1. Modèle RandomForestClassifier :

Un modèle RandomForestClassifier fonctionne en construisant un ensemble d'arbres de décision sur des sous-échantillons aléatoires des données, puis en combinant leurs prédictions par vote majoritaire pour classer les nouvelles observations.

Les forêts aléatoires offrent plusieurs avantages par rapport à d'autres modèles de classification :

- Elles peuvent gérer plusieurs classes, idéal pour notre modèle de classification multi-classe
- Elles fonctionnent bien avec des données volumineuses
- Elles sont efficaces pour repérer les valeurs anormales
- En général, elles évitent le surajustement
- Elles n'ont pas besoin de validation croisée grâce à leurs échantillons "Out of bag".

▪ Résultats avec le modèle RandomForestClassifier :

Rapport de classification (Classification Report)

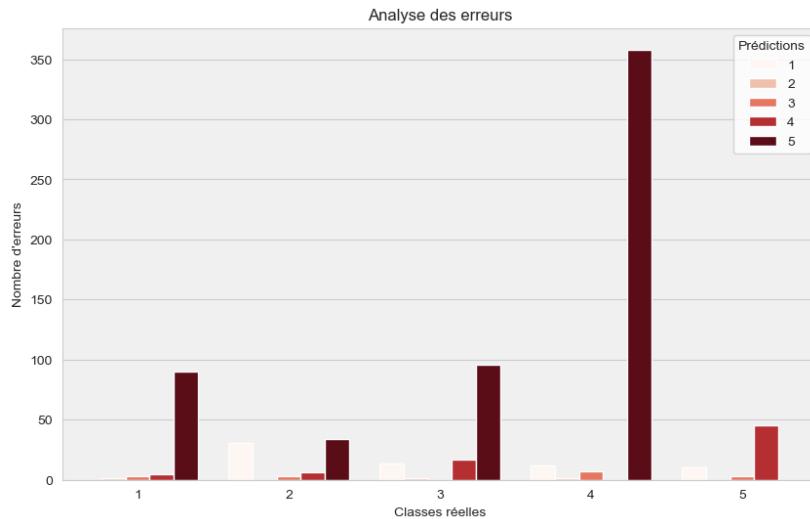
Matrice de confusion

Analyse des erreurs

Classification Report sur l'ensemble de test:					
	precision	recall	f1-score	support	
1	0.68	0.58	0.63	266	
2	0.25	0.01	0.03	70	
3	0.40	0.05	0.08	132	
4	0.40	0.08	0.13	447	
5	0.79	0.98	0.88	2442	
accuracy			0.77	3357	
macro avg	0.50	0.34	0.35	3357	
weighted avg	0.71	0.77	0.71	3357	
Accuracy sur l'ensemble de test: 0.77390527256479					
Classe prédictive	1	2	3	4	5
Classe réelle					
1	155	2	4	8	97
2	23	1	1	2	43
3	27	1	6	15	83
4	13	0	1	36	397
5	10	0	3	29	2400

Analyse des erreurs

Index	classe_reelle	Note_Client_Original	Classe_prédite	diff	Lemmes	Nombre_de_phrases	Classe_longueur_mot	Classe_pt_exclam	Classe_sensiment	Emoticones	Sentimentfr
10324	5	5	5	0	["bienréception", "expéditive"]	1	Très court	E0	Positif	Neutre	0.01
9382	5	5	5	0	["achat", "imprimante", "viens", "recevoir", "permis", "constater", "bonne", "qualité", "service", "m"]	1	moyen	E0	Positif	Neutre	0.37666666666666666666
30764	4	4	5	-1	["commande", "sans", "probl", "matériel", "semble", "état", "voir", "suite", "dysfonctionnement"]	4	moyen	E0	Positif	Neutre	0.4633333333333334
33065	4	4	5	-1	["commande", "faite", "fils", "content"]	2	Court	E0	Positif	Neutre	1.0
15418	4	4	5	-1	["rien", "signaler", "marche", "bien", "macbook"]	2	Court	E0	Neutre	Neutre	0.0
7793	5	5	5	0	["matériel", "qualitéexpédition", "rapide", "soigné"]	1	Très court	E0	Positif	Neutre	0.2466666666666666
14809	5	5	5	0	["livraison", "rapide"]	1	Très court	E0	Positif	Neutre	0.7
31354	5	5	5	0	["sans", "probl", "merapide", "efficace"]	1	Très court	E0	Positif	Neutre	0.4
18694	4	4	5	1	["surprise", "prix", "intéressantdisque", "durs", "conform", "conformes", "description", "compét"]	1	moyen	E0	Positif	Neutre	0.21
24002	5	5	5	0	["service", "collis", "bien", "protégé", "rapidement"]	2	Court	E0	Positif	Neutre	0.3366666666666666
11252	5	5	5	0	["livraison", "rapide", "parfait", "état", "marche", "commande", "matériel", "correspond", "parfait"]	2	moyen	E0	Positif	Neutre	0.45375
2344	5	5	5	0	["équipe", "pros", "pro", "réponde", "information", "technique", "commande", "expédier", "moins"]	1	moyen	E0	Positif	Neutre	0.125
26141	5	5	5	0	["excellent", "livraison", "rapide", "super", "état"]	1	Très court	E0	Positif	Neutre	0.5166666666666666
401	5	5	5	0	["monter", "premier", "matériel", "satisfait", "machine", "fonctionne", "parfaitemment", "encore"]	3	long	E0	Positif	Neutre	0.3196428571428571



▪ Analyse des résultats avec le Modèle RandomForestClassifier :

Analyse globale :

Le modèle présente une grande précision, avec une accuray a 77.3 % c'est-à-dire 77.3% des prédictions sont correctes sur l'ensemble de test.

Analyse par classe :

- Classe 1 (note1) :

La classe 1, qui correspond à la note 1, affiche des scores faibles, avec une précision et un recall proches de 0.68. Le modèle commet 40% des erreurs dont 36% des notes réelles 1 sont prédites en 5.

- Classes 2 (notes 2) :

Les résultats pour la classe 2, correspondant à la note 2, sont insatisfaisants, avec une précision à 0,25 et un recall de 0.01, ce qui se traduit par un F1-score très faible à 0.03. Le

modèle ne parvient pas à prédire cette classe, avec seulement 1 % des prédictions correctement classées selon la matrice de confusion.

- Classe 3 et 4 (note 3 et 4) :

Tout comme la classe 2, le modèle montre une faible précision et un rappel très limité pour les classes 3 et 4. Les erreurs de prédiction sont principalement observées en lien avec la classe 5.

- Classe 5 (note 5) :

C'est la classe pour laquelle le modèle obtient les meilleures performances, avec une précision de 0,79 et un rappel élevé de 0,98.

Conclusion :

Dans l'ensemble, le modèle RandomForestClassifier montre des performances médiocres, étant très efficace pour la classe 5, mais rencontrant de grandes difficultés à prédire la note 2.

9.2.2. Modèle LightGBM Classifier

▪ Résultats :

Rapport de classification (Classification Report)

Matrice de confusion

Classification Report sur l'ensemble de test:					
	precision	recall	f1-score	support	
1	0.69	0.64	0.66	266	
2	0.15	0.07	0.10	70	
3	0.28	0.16	0.20	132	
4	0.53	0.25	0.34	447	
5	0.84	0.96	0.90	2442	
accuracy			0.79	3357	
macro avg	0.50	0.42	0.44	3357	
weighted avg	0.75	0.79	0.76	3357	
Accuracy sur l'ensemble de test: 0.793267798629729					
Classe prédictive	1	2	3	4	5
Classe réelle					
1	171	14	10	14	57
2	24	5	8	9	24
3	28	3	21	24	56
4	14	3	21	110	299
5	12	9	15	50	2356

▪ **Analyse des résultats avec le Modèle LightGBM Classifier :**

Analyse globale :

Accuracy : Le modèle est globalement précis, classant correctement environ 79% des exemples sur l'ensemble de test.

Precision et Recall : La précision est en moyenne de 50% ce qui est relativement faible. Les classes 2 et 3 sont particulièrement mal classées. Le rappel est également faible avec 42%, le modèle a un peu de mal à identifier toutes les classes.

F1-Score : Nous avons un résultat assez faible de 0.44, qui reflète les performances faibles sur l'ensemble du modèle.

Analyse des classes :

- Classe 1 : Précision : 0.69, Rappel : 0.64, F1-score : 0.66

Nous avons des performances moyennes avec 69% des prédictions pour la classe 1 qui sont correctes, et 64% des données réelles de cette classe sont bien prédites.

- Classe 2 : Précision : 0.15, Rappel : 0.07, F1-score : 0.10

Nous avons de mauvais résultats avec 15% des prédictions de cette classe qui sont correctes, et seulement 7% des données réelles qui sont bien prédites. Le faible résultat de la F1-score montre les difficultés du modèle pour cette classe.

- Classe 3 : Précision : 0.28, Rappel : 0.16, F1-score : 0.20

Nous avons seulement 28% des prédictions de la classe qui sont correctes et 16% des données réelles qui sont bien prédites. Nous notons un faible score de 0.20

- Classe 4 : Précision : 0.28, Rappel : 0.16, F1-score : 0.20

Nous avons seulement 28% des prédictions de la classe qui sont correctes et 16% des données réelles qui sont bien prédites. Nous notons un faible score de 0.2.

- Classe 5 : Précision : 0.84, Rappel : 0.96, F1-score : 0.90

Nous avons 84% des prédictions qui sont correctes et 96% des exemples réels qui ont été bien identifiés. Le F1 score de 0.90 montre que le modèle est très performant pour cette classe.

Analyse de la matrice de confusion :

Nous remarquons grâce à la matrice de confusion, qu'il existe des confusions entre les classes 2, 3 et 4. Ces dernières ont un faible F1 score et rappel car beaucoup d'exemples ont été mal classés, cela signifie que le modèle a du mal à prédire ces classes et n'arrive pas à distinguer exactement les différences entre ces classes.

Concernant la classe 5, il s'agit de la classe majoritaire avec 2442 exemples, nous avons un bon résultat sur cette classe avec des métriques assez élevées.

Conclusion

Le modèle LightGBM a bien performé pour la classe 5, mais il montre des faiblesses pour les autres classes, notamment les classes 2, 3 et 4.

Le déséquilibre des classes est probablement la principale cause de ces résultats car la classe 5 représente la grande majorité des exemples et le modèle semble être biaisé vers cette classe, d'où les bons résultats de la classe 5 et les mauvaises performances du modèle sur les autres classes.

9.2.3. Modèle de régression logistique

- **Résultats :**

Rapport de classification (Classification Report)

Matrice de confusion

Dimensions des données combinées (X_test_combined): (3357, 19253)				
Classification Report sur l'ensemble de test:				
	precision	recall	f1-score	support
1	0.60	0.54	0.57	266
2	0.12	0.10	0.11	70
3	0.14	0.13	0.14	132
4	0.30	0.28	0.29	447
5	0.84	0.87	0.86	2442
accuracy			0.72	3357
macro avg	0.40	0.38	0.39	3357
weighted avg	0.71	0.72	0.72	3357
Accuracy sur l'ensemble de test: 0.7223711647304141				

Accuracy sur l'ensemble de test: 0.7223711647304141				
Matrice de confusion:				
	Classe prédictive 1	Classe prédictive 2	Classe prédictive 3	\
Classe réelle 1	144	19	19	
Classe réelle 2	20	7	7	
Classe réelle 3	22	12	17	
Classe réelle 4	23	8	34	
Classe réelle 5	30	14	41	
	Classe prédictive 4	Classe prédictive 5		\
Classe réelle 1	20	64		
Classe réelle 2	9	27		
Classe réelle 3	32	49		
Classe réelle 4	125	257		
Classe réelle 5	225	2132		

▪ Analyse des résultats avec le modèle régression logistique

Analyse globale :

Le modèle atteint une accuracy de 72,2 % sur l'ensemble de test, ce qui semble raisonnable. Cependant, cette valeur est biaisée par la classe majoritaire (classe 5), qui représente une proportion très importante des données (2442 sur 3357 échantillons). La forte dominance de la classe 5 biaise le modèle et limite sa capacité à bien classifier les classes moins représentées.

Le Macro-average : F1-score = 0.39 (faible) indique que les performances du modèle sont médiocres lorsque toutes les classes sont considérées de manière égale, notamment à cause des classes minoritaires.

Le Weighted-average (pondéré) : F1-score = 0.72 (élevé) est gonflé par les bonnes performances sur la classe dominante, ce qui masque les problèmes des classes minoritaires.

Analyse par classe :

- Classes minoritaires (classes 2, 3, et 4) :

Ces classes présentent des précisions et rappels très faibles, avec des F1-scores compris entre 0.11 et 0.29. Cela montre que : Le modèle peine à identifier les échantillons appartenant à ces classes.

Ces classes sont souvent confondues avec la classe majoritaire ou entre elles.

La classe 2 n'est correctement prédite que 7 fois sur 70 (recall de 10 %). Cela rend son prédicteur presque inutile dans ce contexte.

- Classes 1 :

Cette classe modérément représentée obtient des performances intermédiaires (F1-score = 0.57). Les erreurs se répartissent entre les autres classes, notamment la classe 5.

- Classe dominante (classe 5) :

Précision, rappel et F1-score très élevés (respectivement 0.84, 0.87 et 0.86). Cette classe est très bien prédite, avec 2132 des 2442 échantillons correctement classés. Cependant, cette classe dominante absorbe une grande partie des échantillons des autres classes, ce qui amplifie les erreurs pour celles-ci.

Analyse de la matrice de confusion avec le modèle régression logistique :

La matrice de confusion met en évidence des patterns d'erreurs récurrents : La classe 5 capte une grande partie des échantillons des autres classes (par exemple, 225 de la classe 4, 64 de la classe 1, etc.). La classe 3 est souvent confondue avec les classes 4 et 5. Cela indique un chevauchement potentiel dans les caractéristiques utilisées par le modèle.

Conclusion :

Bien que l'accuracy globale soit correcte (72,2 %), le modèle actuel ne parvient pas à gérer efficacement les classes minoritaires.

La classe majoritaire (classe 5) est très bien gérée par le modèle. L'accuracy globale est élevée, ce qui montre que le modèle fonctionne correctement pour une partie importante des données.

Les performances sur les classes 2, 3, et 4 sont très faibles, rendant le modèle inadapté pour les applications nécessitant une bonne prédiction pour toutes les classes.

Certaines classes (3 et 4, par exemple) sont souvent confondues, ce qui indique un manque de discrimination dans l'espace des features.

La prédiction est fortement biaisée vers la classe 5, en partie à cause de sa fréquence élevée.

9.3. Synthèse des résultats des modèles

Modèle	RandomForestClassifier	LightGBM Classifier	régression logistique
Accuracy	0,773	0,793	0,722
precision	0,68	0,69	0,6
recall	0,58	0,64	0,54
F1-score	0,63	0,66	0,57
Erreurs de prediction	36% sont affectés en classe 5 par erreurs		
precision	0,25	0,15	0,12
recall	0,01	0,07	0,1
F1-score	0,03	0,1	0,11
Erreurs de prediction	1% des notes 2 sont bien classées		
precision	0,4	0,28	0,14
recall	0,05	0,16	0,13
F1-score	0,08	0,2	0,14
Erreurs de prediction	95% des notes 3 sont en erreurs de classe		
precision	0,4	0,53	0,3
recall	0,08	0,25	0,28
F1-score	0,13	0,34	0,29
Erreurs de prediction	88% sont prédites en classe 5		
precision	0,79	0,84	0,84
recall	0,98	0,96	0,87
F1-score	0,88	0,9	0,86
Erreurs de prediction	98% des notes 5 sont affectées en classe 5		
%d'erreurs	22,0%		

Ces 3 modèles ne sont pas satisfaisants bien que le LightGBMClassifier se révèle être le modèle le plus performant pour prédire les notes. Il présente des limites en ce qui concerne la prédition classes moins représentées. Nous observons une confusion entre les classes 1 et 5 probablement par la non prise en compte de la négation. De plus, des confusions entre les notes 4 et 5 sont également fréquentes. Cela n'est pas surprenant car les mots liés aux notes 3, 4 et 5 se ressemblent beaucoup, comme le montrent les nuages de mots.

Dans un premier temps, nous essaierons d'améliorer les performances du modèle en intégrant des variables issues du traitement du langage naturel.

Si ces ajustements ne suffisent pas, nous envisagerons de simplifier la variable cible en la réduisant d'abord à trois classes, puis à deux classes, afin de passer d'un modèle multi-classe à un modèle binaire. Nous chercherons ensuite à optimiser ce modèle en ajustant les hyperparamètres.

Enfin, nous explorerons également des modèles de deep learning.

10. Ajustement du modèle

10.1. Intégration de nouvelles caractéristiques NLP ("Natural Language Processing")

10.11. Nouvelle variable intégrant la négation présente dans les données textuelles.

À la suite de l'analyse des erreurs, nous avons observé que la négation et le mot "non" n'étaient pas pris en compte lors de la tokenisation. Or, cette notion est nécessaire pour prédire une note de satisfaction.

Ainsi, nous avons décidé de créer une variable permettant de comptabiliser le nombre de négations présentes dans le titre du commentaire. Après avoir testé ces données en intégrant titre + commentaire, nous nous sommes aperçus que trop de discours, avec ou sans négation, perdent en efficacité pour déterminer si le sens est négatif ou non. C'est pourquoi nous avons choisi de créer une variable uniquement basée sur les titres des commentaires, qui sont plus courts et plus clairs.

- Création d'une variable : « **nbneg** » nombre d'éléments négatifs

10.12. Nouvelle variable de sentiment plus puissante avec BERT.

Nous utilisons le modèle BERT ("nlptown/bert-base-multilingual-uncased-sentiment"), qui s'appuie sur des techniques avancées de traitement du langage pour analyser les sentiments exprimés dans des commentaires multilingues. Ce modèle attribue une note de 1 à 5, ce qui permet de classer les commentaires.

Chaque commentaire, associé à son titre, est alors évalué par ce modèle, qui attribue une nouvelle note de sentiment sur une échelle de 1 à 5.

- Création d'une variable : « **sentiment_dl** » Classes de 1 à 5

10.2. Évaluation des modèles avec l'intégration des nouvelles caractéristiques

10.21. Résultats des modèles

Modèle RandomForest :

Classification Report sur l'ensemble de test:					
	precision	recall	f1-score	support	
1	0.71	0.79	0.75	266	
2	0.50	0.01	0.03	70	
3	0.47	0.06	0.11	132	
4	0.45	0.18	0.26	447	
5	0.83	0.98	0.90	2442	
accuracy			0.80	3357	
macro avg	0.59	0.41	0.41	3357	
weighted avg	0.75	0.80	0.75	3357	
Accuracy sur l'ensemble de test: 0.8004170390229371					
Classe prédictive	1	2	3	4	5
Classe réelle					
1	211	0	4	10	41
2	29	1	3	15	22
3	33	1	8	28	62
4	16	0	1	82	348
5	10	0	1	46	2385

Modèle LightGBM :

Classification Report sur l'ensemble de test:					
	precision	recall	f1-score	support	
1	0.68	0.86	0.76	244	
2	0.29	0.11	0.16	75	
3	0.39	0.28	0.33	130	
4	0.51	0.30	0.38	443	
5	0.89	0.96	0.92	2465	
accuracy			0.82	3357	
macro avg	0.55	0.50	0.51	3357	
weighted avg	0.79	0.82	0.80	3357	
Accuracy sur l'ensemble de test: 0.8200774501042598					
Classe prédictive	1	2	3	4	5
Classe réelle					
1	211	5	13	6	9
2	41	8	16	9	1
3	31	11	37	37	14
4	10	3	21	134	275
5	16	1	9	76	2363

Le modèle le plus performant est toujours LightGBM, avec une amélioration de l'accuracy à 0,82 (contre 0,793) et une nette amélioration dans la réduction des erreurs entre la note 1 et 5, qui est passée de 23 % à seulement 4 %. On constate également une légère amélioration sur l'ensemble des prédictions des autres notes. Nous poursuivons avec l'optimisation des hyperparamètres afin d'améliorer ce modèle.

10.22. Optimisation des hyperparamètres sur le meilleur modèle :

Nous avons réalisé une recherche d'hyperparamètres avec la méthode RandomizedSearchCV afin d'optimiser le modèle.

Voici les meilleurs paramètres identifiés :

- **Num_leaves** : 50 (nombre de feuilles dans l'arbre de décision)
- **min_samples_split** : 2 (nombre minimum d'échantillons requis pour diviser un nœud)
- **n_estimators** : 200 (nombre d'arbres que le modèle va construire)
- **max_depth** : -1 (pas de limite de profondeur pour les arbres)
- **learning_rate** : 0.05 (aider à atteindre une meilleure convergence)

Meilleurs hyperparamètres trouvés : { 'num_leaves': 50, 'n_estimators': 200, 'max_depth': -1, 'learning_rate': 0.05 }																																													
Classification Report sur l'ensemble de test:																																													
<table> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0.69</td> <td>0.86</td> <td>0.77</td> <td>244</td> </tr> <tr> <td>2</td> <td>0.21</td> <td>0.05</td> <td>0.09</td> <td>75</td> </tr> <tr> <td>3</td> <td>0.35</td> <td>0.30</td> <td>0.32</td> <td>130</td> </tr> <tr> <td>4</td> <td>0.51</td> <td>0.30</td> <td>0.38</td> <td>443</td> </tr> <tr> <td>5</td> <td>0.89</td> <td>0.96</td> <td>0.92</td> <td>2465</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.82</td> <td>3357</td> </tr> <tr> <td>macro avg</td> <td>0.53</td> <td>0.50</td> <td>0.50</td> <td>3357</td> </tr> <tr> <td>weighted avg</td> <td>0.79</td> <td>0.82</td> <td>0.80</td> <td>3357</td> </tr> </tbody> </table>		precision	recall	f1-score	support	1	0.69	0.86	0.77	244	2	0.21	0.05	0.09	75	3	0.35	0.30	0.32	130	4	0.51	0.30	0.38	443	5	0.89	0.96	0.92	2465	accuracy			0.82	3357	macro avg	0.53	0.50	0.50	3357	weighted avg	0.79	0.82	0.80	3357
	precision	recall	f1-score	support																																									
1	0.69	0.86	0.77	244																																									
2	0.21	0.05	0.09	75																																									
3	0.35	0.30	0.32	130																																									
4	0.51	0.30	0.38	443																																									
5	0.89	0.96	0.92	2465																																									
accuracy			0.82	3357																																									
macro avg	0.53	0.50	0.50	3357																																									
weighted avg	0.79	0.82	0.80	3357																																									
Accuracy sur l'ensemble de test: 0.8203753351206434																																													
<table> <thead> <tr> <th>Classe prédictive</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> </tr> <tr> <th>Classe réelle</th> <th></th> <th></th> <th></th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>211</td> <td>4</td> <td>14</td> <td>7</td> <td>8</td> </tr> <tr> <td>2</td> <td>41</td> <td>4</td> <td>21</td> <td>8</td> <td>1</td> </tr> <tr> <td>3</td> <td>30</td> <td>9</td> <td>39</td> <td>39</td> <td>13</td> </tr> <tr> <td>4</td> <td>11</td> <td>1</td> <td>29</td> <td>135</td> <td>267</td> </tr> <tr> <td>5</td> <td>13</td> <td>1</td> <td>9</td> <td>77</td> <td>2365</td> </tr> </tbody> </table>	Classe prédictive	1	2	3	4	5	Classe réelle						1	211	4	14	7	8	2	41	4	21	8	1	3	30	9	39	39	13	4	11	1	29	135	267	5	13	1	9	77	2365			
Classe prédictive	1	2	3	4	5																																								
Classe réelle																																													
1	211	4	14	7	8																																								
2	41	4	21	8	1																																								
3	30	9	39	39	13																																								
4	11	1	29	135	267																																								
5	13	1	9	77	2365																																								

Scores de validation croisée : [0.65040321 0.96743927 0.97727849 0.97885074 0.97854643]
Moyenne des scores de validation croisée : 0.9105036263123193

Conclusion :

Le modèle a réussi à classer correctement 82 % des échantillons, mais la sur-représentation de la classe 5 fausse les résultats. En effet, sur les classes sous-représentées, les précisions ne sont que de 0,21 pour la classe 2 et de 0,35 pour la classe 3. La matrice de confusion confirme ce déséquilibre de classe et affiche des confusions principales entre les classes 1, 2 et 4. Bien que le modèle fonctionne mieux pour les classes 1 et 5, il présente toujours des lacunes pour les classes 2 et 3. La moyenne des scores de validation croisée est de 0.91 ce qui indique que le modèle a une performance plutôt satisfaisante.

Malgré l'application de techniques comme le suréchantillonnage des classes minoritaires et l'ajustement des poids des classes, le modèle n'est pas suffisamment robuste pour prédire les

cinq classes de manière fiable. Par conséquent, nous envisageons de regrouper certaines classes afin de réduire ce déséquilibre et d'améliorer la solidité du modèle.

10.2. Évaluation du modèle exclusivement sur les métadonnées

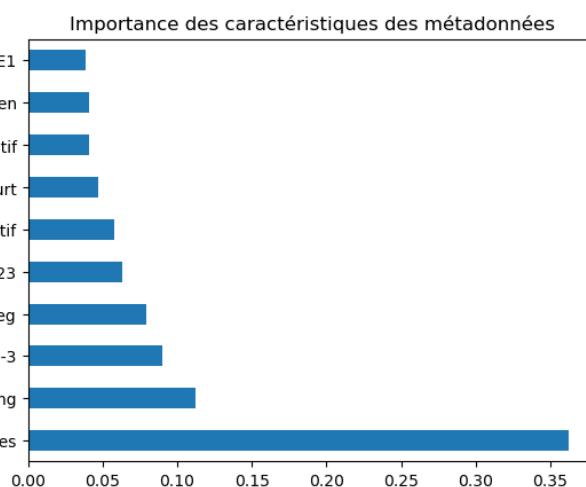
Nous cherchons à comprendre l'impact des données sur les résultats du modèle. On appelle ici, métadonnées l'ensemble des variables sans les données textuelles (commentaires et titres de commentaires).

Nous allons analyser ces métadonnées sur un modèle de RandomForest à travers l'analyse des performances et l'importance des caractéristiques.

Classification Report sur l'ensemble de test:					
	precision	recall	f1-score	support	
1	0.67	0.76	0.71	266	
2	0.13	0.16	0.14	70	
3	0.27	0.39	0.32	132	
4	0.34	0.44	0.38	447	
5	0.92	0.83	0.87	2442	
accuracy			0.74	3357	
macro avg	0.47	0.52	0.49	3357	
weighted avg	0.78	0.74	0.76	3357	

Accuracy sur l'ensemble de test: 0.7402442657134346

Classe prédictive	1	2	3	4	5
Classe réelle					
1	202	33	15	6	10
2	32	11	15	9	3
3	23	21	52	29	7
4	18	11	70	197	151
5	27	8	39	345	2023



❖ Analyse du rapport de classification :

Précision, rappel et F1-score par classe :

Classe 1 : Précision : 0,67 // Rappel : 0,76 // F1-score : 0,71

Parmi les prédictions de la classe 1, 67 % étaient correctes. Le modèle a identifié correctement 76 % des instances réelles de cette classe. Une performance globale assez moyenne pour cette classe.

Classe 2 : Précision : 0,33 // Rappel : 0,16 // F1-score : 0,21. On note ici une faible performance, suggérant des difficultés du modèle à identifier ou à prédire cette classe.

Classe 3 : Précision : 0,27 // Rappel : 0,39 // F1-score : 0,32. La performance encore plus faible probablement due à une confusion de cette classe avec d'autres.

Classe 4 : Précision : 0,34 // Rappel : 0,44 // F1-score : 0,38. Cela indique également une difficulté à bien classifier cette classe.

Classe 5 : Précision : 0,92 // Rappel : 0,83 // F1-score de 0,87. Le modèle atteint une excellente performance particulièrement cette classe.

❖ **Moyennes globales :**

Accuracy : Le modèle est correct dans 74 % des cas.

Macro average : Précision (0,47) , rappel(0,52) et F1-score(0,49). Cette moyenne simple sur toutes les classes montre que les performances sont très déséquilibrées entre les classes.

Weighted average : Précision (0,78) , Rappel(0,74) et F1-score(0,76). Cette moyenne pondérée confirme que le modèle performe mieux sur les classes majoritaires, comme la classe 5.

❖ **Analyse du tableau de confusion :**

Classe 1 : 202 instances sont bien classées et 33, 15, 6 et 10 instances respectivement mal classées comme classe 2, 3, 4 et 5. Le modèle confond parfois la classe 1 avec les autres.

Classe 2 : Seulement 11 instances sont correctement classées contre 32 mal classées comme classe 1, et 15, 9, 3 mal classées comme d'autres classes. Cela confirme la faible performance pour cette classe.

Classe 3 : 21 bien classées, mais 23 et 51 instances respectivement mal classées comme classe 1 et classe 5. Les erreurs avec la classe 5 sont particulièrement nombreuses.

Classe 4 : Seulement 197 correctement classées, avec beaucoup d'erreurs vers la classe 5 (151 instances). Cela indique que le modèle a des difficultés à distinguer entre les classes 4 et 5.

Classe 5 : Excellente classification : 2023 instances correctement identifiées, bien au-dessus des autres classes.

Conclusion :

L'application des métadonnées montre une performance globale satisfaisante avec un taux d'accuracy de 74 %. Cependant, on constate une grande disparité entre les classes. Les classes majoritaires, telles que la classe 5, affichent de meilleures performances.

Cette analyse nous permet de conclure que les métadonnées jouent un rôle important dans la capacité du modèle à classifier les données. Nous observons également que la caractéristique "nombre de phrases" est un facteur déterminant, suivie par la taille du texte et le nombre d'avis du client.

10.3. Classification de la variable cible en trois catégories

Nous tentons de classer la variable "note" en trois catégories, en nous basant sur les résultats des premiers tests de modélisation et sur l'idée de qualité.

Cette classification s'aligne avec la sémantique de la satisfaction client, qui regroupe souvent les avis en trois ensembles, notamment dans le cadre des enquêtes NPS (Net Promoter Score)

- Promoteurs (score 9-10) : Clients très satisfaits
- Passifs (score 7-8) : Clients satisfaits, mais sans enthousiasme marqué
- Détracteurs (score 0-6) : Clients

Cette approche nous permet donc d'avoir une vision plus claire et efficace de la satisfaction client.

- **Création d'une variable :** « Class_note » : classification de la note client
 - Note de 4-5 = "Promoteur",
 - Note 3 = "Passif"
 - Note 1-2 = "Détracteur"
- **Impact sur le modèle RandomForestClassifier:**

Classification Report sur l'ensemble de test:				
	precision	recall	f1-score	support
1	0.81	0.74	0.77	336
2	0.17	0.01	0.01	132
3	0.94	0.99	0.97	2889
accuracy			0.93	3357
macro avg	0.64	0.58	0.58	3357
weighted avg	0.90	0.93	0.91	3357
Accuracy sur l'ensemble de test: 0.9273160560023831				
Classe prédictive	1	2	3	
Classe réelle				
1	248	4	84	
2	36	1	95	
3	24	1	2864	

Conclusion :

Le rapport de classification indique que le modèle a une précision élevée (0.94) pour la classe 3 (note 4 et 5) et un bon taux de rappel (0.99). Mais les performances pour la classe 2 (note 3) sont très faibles, avec une précision de seulement 0.17 et un rappel de 0.01. Ces résultats montrent un déséquilibre trop important dans la classification des classes, particulièrement pour la classe 2 (note 3).

Cette classification de note n'est donc pas retenue.

10.4. Classification de la variable cible en variable binaire

Nous allons donc transformer la variable cible en un modèle binaire, où la valeur 1 correspond aux notes 1 et 2, et la valeur 2 aux notes 3, 4 et 5. Nous espérons que cette modification permettra au modèle de mieux classifier les notes 2 et 3, qui posent actuellement problème.

Nous avons opté pour cette répartition car, comme nous l'avons noté avec les nuages de mots, les notes 3, 4 et 5 sont particulièrement difficiles à distinguer à partir des mots exprimés.

10.4.1. Résultats des modèles

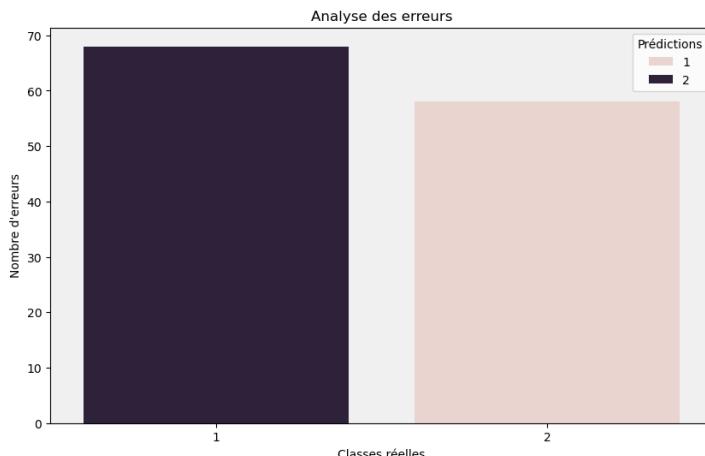
Modèle de classification RandomForestClassifier avec classe_note

Classification Report sur l'ensemble de test:				
	precision	recall	f1-score	support
1	0.85	0.68	0.76	319
2	0.97	0.99	0.98	3038
accuracy			0.96	3357
macro avg		0.91	0.83	0.87
weighted avg		0.96	0.96	0.96
Accuracy sur l'ensemble de test: 0.958593982722669				
Classe prédictive	1	2		
Classe réelle				
1	217	102		
2	37	3001		

Modèle de classification Modèle LightGBM avec classe_note

Classification Report sur l'ensemble de test:				
	precision	recall	f1-score	support
1	0.82	0.78	0.80	319
2	0.98	0.98	0.98	3038
accuracy			0.96	3357
macro avg		0.90	0.88	0.89
weighted avg		0.96	0.96	0.96
Accuracy sur l'ensemble de test: 0.9624664879356568				
Classe prédictive	1	2		
Classe réelle				
1	249	70		
2	56	2982		

Analyse des erreurs



classe réelle	Classe prédite	Note_Client_Original	Lemmes	Lemmes_titre_commentaire	Nombre_de_phrases	Classe_longueur_mot	Classe_pt_exclam	Classe_sentiment	Emoticones	Sentimentfr	nombre neg	negatif	sentiment
2	2	2	['matériel', 'conforme', 'commande', 'service'...]	[]	1	Court	E0	Positif	Neutre	0.328000	0	0	5
1	1	1	['voil', 'bient', 'mois', 'commande', 'apple'...]	['deuxi', 'commande', 'toujours', 'colis']	4	long	E3	Positif	Neutre	0.029792	1	1	1
1	1	1	['achat', 'effectué', 'jours', 'livraison', 'p...']	['éviter']	5	long	E1	Négatif	Neutre	-0.050000	0	0	1
2	2	2	['procédure', 'remplacement', 'identique', 'pu...']	['procédure', 'remplacement']	1	long	E0	Neutre	Neutre	0.000000	0	0	3
2	2	2	['site', 'clair', 'données', 'compl', 'achat'...]	['site', 'clair', 'données', 'compl', 'pour']	1	moyen	E0	Positif	Neutre	0.270000	0	0	5

En réduisant la variable "note client" à une représentation binaire, nous observons une amélioration significative des performances du modèle.

Le modèle LightGBM s'avère le plus efficace, atteignant une précision de 0.962. Cette simplification permet de surmonter les difficultés de prédiction de la note 2.

10.42. Optimisation des hyperparamètres sur le meilleur modèle

Nous réalisons une recherche d'hyperparamètres avec la méthode RandomizedSearchCV afin d'optimiser le modèle.

Voici les meilleurs paramètres identifiés :

- **Num_leaves** : 50 (nombre de feuilles dans l'arbre de décision)
- **min_samples_split** : 2 (nombre minimum d'échantillons requis pour diviser un nœud)
- **n_estimators** : 200 (nombre d'arbres que le modèle va construire)
- **max_depth** : -1 (pas de limite de profondeur pour les arbres)
- **learning_rate** : 0.05 (aider à atteindre une meilleure convergence)

```

Meilleurs hyperparamètres trouvés : {'num_leaves': 50, 'n_estimators': 200, 'max_depth': -1, 'learning_rate': 0.05}
Classification Report sur l'ensemble de test:
precision    recall   f1-score   support

          1       0.81      0.79      0.80      319
          2       0.98      0.98      0.98     3038

   accuracy                           0.96      3357
macro avg       0.89      0.88      0.89      3357
weighted avg    0.96      0.96      0.96      3357

Accuracy sur l'ensemble de test: 0.9624664879356568

Classe prédictive 1 2
Classe réelle
 1 251 68
 2 58 2980

```

L'accuracy globale du modèle sur l'ensemble de test est de 96.25%, ce qui indique une excellente performance générale. Cela signifie que le modèle a correctement classé 96.25% des échantillons testés.

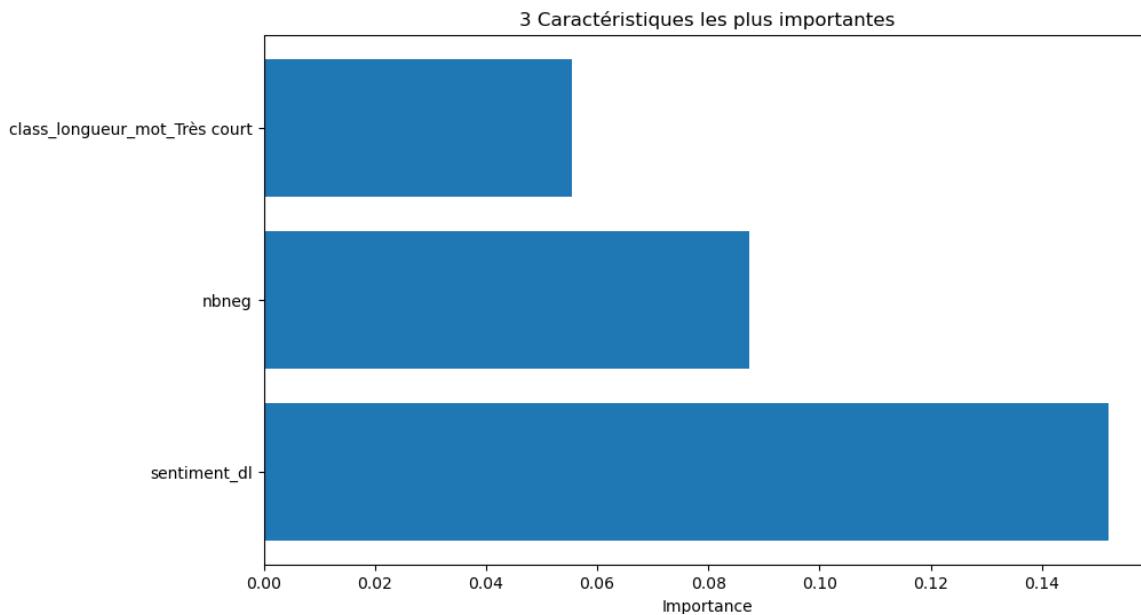
Conclusion :

Globalement, le modèle affiche de bonnes performances, mais il est important de rester prudent face à ces résultats prometteurs car la classe 1 est sous-représentée et des confusions persistent avec la classe 2. Il serait nécessaire de réévaluer ce modèle sur un ensemble de données plus équilibré afin de confirmer sa véritable robustesse.

10.43. Interprétabilité du modèle

Comme nous l'avons déjà noté, les métadonnées jouent un rôle clé dans le modèle de classification. Cependant, l'intégration de données issues du traitement du langage naturel (NLP) modifie le comportement du modèle. En effet, la variable dérivée du modèle BERT (bert-base-multilingual), « sentiment_dl » se révèle être la plus impactante pour le modèle, suivie de la variable de négation.

Comme nous l'observons dans le graphique, les trois variables ayant le plus d'influence sur le modèle sont les variables « sentiment_dl », « nbneg » qui représente la négation dans le titre et le nombre de mots dans le commentaire.



11. Modélisation avec deep learning

Nous avons appliqué dans cette partie un modèle de réseau de neurones récurrent avec une couche LSTM (Long Short-Term Memory) implémenté à l'aide de la bibliothèque Keras de TensorFlow.

Les résultats sont présentés ci-après.

```

13425/13425 83s 6ms/step - accuracy: 0.7450 - loss: 0.7406
Epoch 2/5
13425/13425 84s 6ms/step - accuracy: 0.7724 - loss: 0.6386
Epoch 3/5
13425/13425 1191s 89ms/step - accuracy: 0.7858 - loss: 0.5921
Epoch 4/5
13425/13425 145s 11ms/step - accuracy: 0.8034 - loss: 0.5520
Epoch 5/5
13425/13425 182s 14ms/step - accuracy: 0.8125 - loss: 0.5245
210/210 2s 5ms/step - accuracy: 0.7715 - loss: 0.6677
Loss: 0.6841996312141418, Accuracy: 0.7670192122459412
1049/1049 6s 5ms/step

Commentaire Predicted_Note
0 bonjour, j'ai acheté ma première configurzton... 5
1 je partais à la recherche de mon premier ecran... 5
2 j'ai récemment eu un problème avec ma carte gr... 5
3 je passe par materiel.net car historiquement j... 4
4 cela fait des années que j'achète sur votre si... 5
...
33558 commande reçue rapidement, téléphone opération... 5
33559 livraison rapide état du téléphone nickel rasj... 5
33560 téléphone reçu dans des délais très rapides. q... 5
33561 envoi impeccable, délai respecté, produit état... 5
33562 expédition très rapide . le téléphone a l 'air... 4

[33563 rows x 2 columns]

```

- Analyse des résultats par époque de l'ensemble d'entraînement

Epoch 1/5 - Accuracy: 0.7450 - loss: 0.7406 : La précision sur l'ensemble d'entraînement est 74,5% et la perte sur cet ensemble est de 0,74.

Epoch 2/5 - Accuracy: 0.7766 - Loss: 0.619 : La précision sur l'ensemble d'entraînement est 77,6% et la perte sur cet ensemble est de 0,61.

Les autres époques montrent une amélioration continue de la performance :

Epoch 3/5 - Accuracy: 0.7858 - Loss: 0.5921

Epoch 4/5 - Accuracy: 0.8034 - Loss: 0.5520

Epoch 5/5 - Accuracy: 0.8125 - Loss: 0.5245

La précision augmente progressivement (de 74,5 % à 81,2 %), ce qui montre que le modèle apprend efficacement.

La perte diminue légèrement (de 0.74 à 0.52), ce qui indique que le modèle devient meilleur pour prédire les classes correctement.

La perte a continué à diminuer, atteignant 0.5245, ce qui montre que le modèle continue d'apprendre.

- Évaluation du modèle et Interprétation sur l'ensemble de validation / test

210/210 - accuracy: 0.7715 - loss: 0.6677.

Le modèle évalue les performances sur 210 mini-batches (car l'ensemble de test est plus petit que l'entraînement).

La précision et la perte sur l'ensemble de test sont respectivement de 77,15 % et 0,66.

La précision sur l'ensemble de test (77 %) est inférieure à celle de l'entraînement (81 %), ce qui pourrait être dû à un surapprentissage.

La perte sur l'ensemble de test est de 0.6677, ce qui est plus élevé que celle sur l'entraînement. Le modèle fonctionne précisément dans les données d'entraînement mais a du mal à généraliser aux nouvelles données.

- Prédiction sur tous les commentaires.

Une vérification a permis de noter que le modèle prédit relativement bien les commentaires.

A titre d'exemple un commentaire positif comme "très bonne bécane pour mon usage" obtient à une note prédite de 5. Un commentaire négatif comme "mauvaise communication..." est associé à une note de 1.

- Distribution et pertinence des prédictions.

Les résultats montrent que le modèle fonctionne correctement pour détecter le sentiment (positif ou négatif) des commentaires et les associer à une note prédite cohérente. Toutefois, il est possible que le modèle ait encore des erreurs sur des classes déséquilibrées.

- Améliorations possibles :

Nous avons été limités par des contraintes techniques, notamment la puissance de nos ordinateurs, ce qui a ralenti notre capacité à explorer les potentialités des modèles de deep learning, tel que Flaubert.

Pour améliorer les performances et la généralisation du modèle,

On pourrait :

- Procéder à la validation croisée pour une évaluation plus robuste du modèle.
- Équilibrer les classes
- Augmenter la taille de l'ensemble de données (plus de commentaires).
- Ajuster la taille des vecteurs d'embedding (output_dim) ou la dimension LSTM.
- Utiliser une stratégie d'arrêt précoce (early stopping) pour éviter le surapprentissage.
- Augmenter max_len pour capturer plus de contexte des commentaires.

12. Conclusion

L'objectif principal de ce projet était de créer un modèle de machine learning capable d'évaluer et de prédire le niveau de satisfaction des clients à partir de leurs commentaires en attribuant une note de satisfaction sur une échelle de 1 à 5.

Bien que nous ayons rencontré quelques difficultés tout au long de notre projet, notamment le déséquilibre des classes au sein de notre ensemble de données et des contraintes techniques liées à la puissance de nos ordinateurs pour tester des modèles de deep learning, nous avons réussi à identifier des approches prometteuses pour prédire des notes de satisfaction fiables.

Pour répondre aux défis du modèle multi-classes déséquilibré, nous avons mis en place des techniques de suréchantillonnage et d'ajustement des poids des classes. Cependant, notre modèle éprouve encore des difficultés à prédire de manière fiable les cinq catégories de satisfaction, affichant une précision de 0,79 et des problèmes de confusion entre les classes. Après avoir analysé ces erreurs, nous avons réajusté le modèle en intégrant de nouvelles caractéristiques, ce qui a permis d'améliorer la précision à 0,82. Nous avons également constaté que les métadonnées jouent un rôle important dans la performance du modèle. En particulier, la caractéristique "nombre de phrases" apparaît comme un facteur important, suivie par la taille du texte, le nombre d'avis clients et la présence de négations dans le titre. Toutefois, malgré ces bons résultats grâce aux nouvelles caractéristiques, les prédictions restent confuses entre les notes 1 et 2 ainsi qu'entre les notes 4 et 5.

Nous avons alors décidé de privilégier un modèle binaire, moins complexe mais plus robuste face au déséquilibre des classes. Grâce à cette adaptation, nous avons atteint une précision de 0,96 après optimisation des hyperparamètres, ainsi qu'une meilleure affectation des notes.

Bien que ce modèle ne corresponde pas entièrement à notre objectif initial, il offre une solution opérationnelle efficace permettant à l'entreprise de réagir rapidement aux avis négatifs et d'améliorer la satisfaction client.

Pour améliorer les performances du modèle, nous suggérons plusieurs pistes, notamment la réévaluation de l'ensemble de données pour obtenir un équilibre entre les classes, l'exploration de modèles de deep learning et de meilleurs outils informatiques.

Enfin, ce projet nous a permis de développer nos connaissances dans l'analyse des sentiments sur des données déséquilibrées, d'appliquer concrètement les connaissances acquises en cours et d'explorer des solutions pratiques pouvant être appliquées en entreprise pour améliorer la gestion de la satisfaction client.