# CuReAnalytics

Customer’s Reviews Analytics

Octobre à Décembre 2023


Michel Douglas Piamou

Mike Boudhabhay

Edwige Fève

# Sommaire

**Présentation de l'équipe projet:** **6**
**Contexte** **7**
**Enjeux** **8**
- Du point de vue technique : 8
- Du point de vue économique : 9
- Du point de vue scientifique : 9

**Objectifs** **9**
Quels sont les principaux objectifs à atteindre ? 9
**Cadre** **10**
**Source des données:** **11**
**Méthodes explorées:** **11**
**Web Scraping avec BeautifulSoup:** **12**
**Exploration des Données :** **13**
Préparation des données: 14
Enrichissement du jeu de données : 14
**Data Visualisation** **15**
Analyse de la distribution des variables: 15
Graphique 1 : Proportion de reviews par rating: 15
Graphique 2: Nb de commentaires en fonction du jour de la semaine : 16
Graphique 3: Longueur moyenne du commentaire =f(rating) 16
Graphique 4: Boîte à moustaches pour vérifier les outliers et les médianes: 18
Graphique 5 : Word cloud sur les commentaires 19
Analyse des Réponses fournisseurs : 20
Graphique 6 : Nombre de réponses fournisseur par jour : 20
Graphique 7: Distribution des délais de réponse des fournisseurs : 21
Graphique 8: Analyse du délai moyen par Rating: 21
Graphique 9: Evaluation moyenne et nombre de réponses fournisseurs par mois: 22
Analyse de corrélation : 23
Graphique 10 : Analyse de corrélation par heatmap : 23
Tests statistiques: 24
**Synthèse de l'exploration des données et de leur visualisation:** **25**
Prédiction de la note du client en fonction des commentaires 27
Problématique 27
Démarche 28
Métriques de performance utilisées 28

Modèles de référence dit Baseline models: 29
Modèles testés 29
Standardisation des données 29
Sous-échantillonage aléatoire 29
Choix du modèle et Optimisation 29
Résultats pour la classification binaire 30
Graphique 11 - Radar de comparaison des métriques 30
Graphique 12 - Précision dans la prédiction du sentiment négatif 31
Graphique 13 - Temps de d'entraînement vs Temps de prédiction 32
Les modèles les plus coûteux en temps de prédictions sont SVC, KNN, Forêt Aléatoire. 32
Les modèles les plus coûteux en temps d'entraînement sont SVC, Forêt aléatoire(r_fo) et Gradient Boost (gdb). 32
Graphique 14 - Comparaison sur 5 métriques 32
Graphique 15 - métriques top 4 33
Conclusion sur la modèle de base de classification binaire 33
Modèles "Bag of Words" avec Tf-Idf ET CountVectorize 34
Démarche 34
Résultats pour TF-IDF Vectorize 36
Graphique 16 - radar des métriques des modèles 36
Graphique 17 - Précision pour la classe des sentiments négatifs 37
Graphique 18 - temps d'entraînement vs temps de prédiction 37
Graphique 18 - Comparaison pour 5 métriques 38
Graphique 19 - Métriques du top 4 39
Graphique 20 - Radar des métriques du top4 40
Graphique 21 - Métriques du modèle LinearSVC 40
Conclusion sur le Bag of Words avec Tf-IdF 41
Résultats pour Count Vectorize 41
Graphique 22 - Comparaison des métriques et temps 41
Graphique 24 - Précision globale des modèles 42
Graphique 25 - Métrique du top 3 43
Graphique 25 - radar top 3 des modèles 43
Conclusion sur le Bag of Words avec CountVectorizer 43
Résultats de l'optimisation avec Grid Search CV 44
Graphique 26 - Radar - Comparaison avec GridSearchCV 45
Graphique 27 - Barplot - Comparaison avec GridSeachCV 45
Graphique 28 - Group Barplot - Comparaisons avec sans GridSearchCV 46
Application d'un réseau dense de neurones 46
Résultats du modèle Tf Idf & Réseau dense de neurones 47

Graphique 29 - Réseau de Neurone Dense vs Régression Logistique 47
Graphique 30 - Barplot - Réseau de neurones dense vs Régression Logistique 47
Modèle de réseau de neurones - GRU : Gated Recurrent Units 48
Qu'est ce qu'un Réseau de Neurones Récurrent? 48
Qu'est ce qu'un GRU? 48
Démarche de mise en oeuvre d'un modèle GRU 48
Graphique 31 - Modèle GRU_1 49
Résultats des modèles GRU 50
Graphique 32 - radar des performance des modèles GRU 50
Graphique 33 - Group barplot - Performance des modèles GRU 51
Classification multi-classes 51
Figure 34 - Métriques-macro des modèles 52
Figure 35 - Barplot Métriques -macro des modèles 52
Analyse de résultats 52
Figure 36 -Rapport de classification et Matrice de confusion régression logistique baseline 53
Figure 37 -Rapport de classification et Matrice de confusion régression logistique baseline 53
Figure 38 - Modèle GRU_mcl0: matrice confusion et rapport de classification 54
Comparaison Modèle de base versus Bags of Words versus GRU 54
Graphique 39 - Radar - Comparaison des métriques 55
Graphique 40 - barplot - Comparaison des métriques 55
En conclusion, 55
Analyse de sentiments 57
Preprocessing & Features engineering: 57
Méthodes étudiées : 58
Prédiction de la réponse fournisseur en fonction de la note du client: 65
Tests statistiques 65
1/ Homogénéité des variances / Test LEVENE 65
Output 65
2/ Normalité des résidus 66
Graphique 66
Test de KRUSKAL-WALLIS : 66
Output : 66
Test de DUNN 67
Graphique 67
Synthèse des tests statistiques 68
Premières modélisations avec analyse de sentiment 68
Démarche 68

1/ MODELISATION TEXT MINING 68
Output MODELISATION TEXT MINING 68
2/ MODELISATION VADER 69
Visualisation de la relation sentiment pour « pos », « neu », « neg » et rating : 70
3/ MODELISATION BERT : 70
Matrice de confusion Bert: 71
Un retour dans les “Verbatims” 72
Démarche 72
Analyse des réponses pour les ratings de 4 ou plus : 73
Constat 74
Tests complémentaires : 74
Analyse de Sentiment simple "TextBlob" 75
Constat 75
Analyse des Réponses pour les Ratings Entre 1 et 3: 76
En Conclusion: 79
Topic modeling: 80
Création d’un modèle interactif via une Chatbot : 80
Conclusions de l'Étape 3 : 81
**Rapport d’exploration des données : 86**
**Notebooks du projets : 86**

# Executive Summary

Notre projet visait à prédire les notations des clients, catégoriser les commentaires, proposer des réponses automatiques, et analyser les sentiments.

Trois approches successives ont été adoptées dans l'analyse des commentaires clients :

**Modèles de classification :** l'utilisation de divers modèles pour prédire le sentiment client en se basant sur la longueur du commentaire, du titre, et la durée depuis la transaction. La régression logistique simple a montré des performances notables.

**Bag of Words avec TF IDF et Count Vectorizer :** l'application du Bag of Words avec TF IDF et Count Vectorizer a permis une bonne précision d'ensemble. Cependant, des limitations subsistent dans la précision de prédiction du sentiment négatif.

**Réseaux de Neurones Récurrents GRU :** Les GRU ont surpassé les modèles précédents, atteignant des précisions globales dépassant 91% améliorant nettement la précision dans la prédiction du sentiment négatif.

Le **rééchantillonnage et la prise en compte du poids des classes** pour les réseaux de neurones a été essentiel pour équilibrer les données quel que soit le modèle.

Pour la prédiction du retour fournisseur, une approche parallèle a été adoptée, mettant l'accent sur le feature engineering, la recherche d'une trame standard, et le fine-tuning des modèles. **L'analyse des commentaires fournisseur a révélé des tendances spécifiques telles qu'une standardisation des réponses, confirmant nos hypothèses.**

L'**analyse des sentiments**, allant des word clouds à BERT, a révélé des nuances. Les approches supervisées et semi-supervisées, notamment Naïve Bayes et le Topic Modeling, ont fourni des informations tangibles.

La **mise en place d'un Chatbot interactif**, intégrant une évaluation des sentiments et présentant les quatre principaux sujets sous-jacents, a rendu conviviale l'analyse des commentaires. Les résultats concluants mettent en lumière l'utilité de cet outil.

En résumé, **nos analyses approfondies nous ont permis d'élaborer un bon système de prédictions de l'évaluation des commentaires**, mais comme dans tout modèle de prédiction, des efforts continus sont nécessaires pour le perfectionner et optimiser les paramètres. À un stade avancé, une modélisation de sujets plus précis pourrait diriger les commentaires vers les départements spécifiquement concernés.

# Introduction au projet

## Présentation de l'équipe projet:

**Pour chacun des membres du groupe, préciser le niveau d'expertise autour de la problématique adressée ?**

- Michel : Mon cœur de métier est le Conseil en transformation SI. sensibilisé aux enjeux de performance de la chaîne de production vis-à-vis de la satisfaction client.
- Mike : Mon background est orienté sur l'entrepreneuriat ( Gérant PME-TPE) avec un socle de connaissances financières, contrôle de gestion.Pas de spécialisation antérieures sur la Supply chain.
- Edwige : mon cœur de métier est la Supply Chain, par contre le scraping de données et la catégorisation de commentaires complètement nouveaux pour moi.

**Êtes-vous entré en contact avec des experts métiers pour affiner la problématique et les modèles sous-jacents ? Si oui, détaillez l'apport de ces intéractions.**

Pas pour le moment. A confirmer si l'on veut intégrer une approche plus commerciale/marketing à notre analyse.

**Avez-vous connaissance d'un projet similaire au sein de votre entreprise, ou bien dans votre entourage ? Quel est son état d'avancement ? En quoi vous a-t-il aidé dans la réalisation de votre projet ? En quoi votre projet contribue-t-il à l'améliorer ?**

- Michel: non
- Mike: non
- Edwige : non

# Contexte

**Contexte d'insertion du projet dans notre métier :**

La satisfaction du client, la réputation en ligne et l'ajustement aux besoins du marché sont au cœur de la problématique de toute entreprise qui souhaite rester compétitive et profitable sur le long terme. Préoccupation d'autant plus cruciale que nous sommes dans un monde dématérialisé où l'information est disponible et circule à grande vitesse.

Le développement de nouveaux produits ou la réponse à une demande plus importante que prévue peut engendrer - en interne - de nombreux surcoûts, des retards, des non-conformances qu'il est important de capturer mais aussi d'un point de vue externe - une insatisfaction du client final.

De la même façon ne pas être capable de s'adapter à une baisse de la demande sur un produit qui a été cannibalisé ou qui devient obsolète est critique pour l'entreprise et ses distributeurs afin d'éviter de générer des stocks de futurs invendus.

La latence d'une supply chain aujourd'hui est beaucoup plus longue que ce que l'on pourrait penser. Les flux sont globalisés avec des fournisseurs, des sites de production et des clients très dispersés géographiquement.

Réagir au plus vite et rester connecté à la demande est donc vital pour rationaliser les coûts opérationnels.

Enfin, il est à noter qu'au-delà de la qualité et de la disponibilité d'un produit, les attentes des clients sont aussi environnementales. Pouvoir capturer ces attentes permet aux entreprises de réfléchir à leur positionnement et de retravailler leurs produits.

**En résumé, être à l'écoute des ses clients au-delà d'une simple mesure de taux de service s'avère fondamental pour maintenir sa base de clients, mais en conquérir de nouveaux et ce afin de rester prospère mais aussi de se positionner avantageusement vis à vis de la concurrence.**

# Enjeux

Aujourd'hui, les données de satisfaction clients sous formes de commentaires, d'avis et de notation sont disponibles sur Internet.

Leur exploitation pertinente peut permettre aux entreprises d'identifier leurs domaines d'amélioration potentiels, d'évaluer la satisfaction globale de leurs clients, de communiquer les messages de la marque, d'identifier les produits ayant fait l'objet du plus grand nombre d'avis, et bien d'autres choses encore. De ce fait, l'exploitation de ces commentaires est centrale pour les entreprises d' e-commerce telles que Amazon, eBay, NetFlix, etc.

On note que le coût global de mauvaises expériences clients est estimé en trillions de dollars. (https://blog.hubspot.com/service/customer-service-stats)

Toutes ces informations permettent donc :

- d'évaluer de manière qualitative et quantitative, la performance de leur organisation supply chain et au-delà,
- De mesurer l'adéquation d'un service ou d'un produit par rapport aux attentes du consommateur final et de se réajuster si nécessaire,
- De synthétiser les retours clients pour travailler sur des axes d' amélioration,
- D'automatiser - de manière performante - le prétraitement des retours des clients insatisfaits (réponse adéquate, redirection sur le service concerné).

**Notre projet : Répondre à ces enjeux avec les outils et techniques de la Data Science et être capable de fournir une classification des problématiques mentionnées par les clients.**

- ## Du point de vue technique :

La contrainte technique s'affine au fur et à mesure de l'avancée de notre projet mais d'ores et déjà nous pouvons mentionner les problématiques suivantes :

  - Aura-t-on des limites pour concaténer l'information ( i.e clés communes) ou de langue lorsqu'un site collecte des commentaires en plusieurs langues?
  - Sera-t-on capable d'intégrer la spécificité de chaque site ciblé qui demande d'adapter son programme d'extraction à une nouvelle source d'informations?

- Devrons-nous mettre à jour régulièrement notre programme de scraping si le site évolue en parallèle?
- Pourrons nous accéder aux données qui peuvent nécessiter des identifiants ou qui peuvent être bloquées car les sites se protègent des requêtes en masse que génèrent les programmes de scraping.

### Du point de vue économique :

Avec les médias sociaux et l'hyper connectivité aujourd'hui, les entreprises ne peuvent plus être auto-centrées sur leurs processus internes uniquement mais être à l'écoute du marché pour :

- comprendre les nouvelles tendances,
- anticiper des changements de comportement clients,
- surveiller leur notoriété, la réputation ayant un impact direct et quasiment immédiat sur le chiffre d'affaires,
- utiliser la disponibilité des retours clients comme un levier si l'information est exploitée correctement.

Ce travail de veille et d'analyse peut être extrêmement bénéfique si l'entreprise est assez agile pour y répondre ou catastrophique si elle ne sait pas capter l'évolution de son marché.

### Du point de vue scientifique :

Nous allons explorer des techniques de NLP ( Natural Language Processing), d'intelligence artificielle et de deep learning pour le traitement du texte des commentaires clients.

## Objectifs

### Quels sont les principaux objectifs à atteindre ?

- **Être capable de webscraper des données pour extraire des commentaires.**
- **Prédire la satisfaction client** ( note / nombre d'étoiles) à partir des commentaires.
- **De ces commentaires, identifier les catégories de sujets problématiques** : a-t on des problèmes de livraison et de retard, des produits qui arrivent abîmés, des produits potentiellement périmés, etc. )
- A partir des commentaires clients, **être capable d'automatiser une réponse**

- **Détection du sentiment client** : positif, neutre ou négatif - à confirmer
- Création d'une dashboard de suivi - à confirmer

## Cadre

Collecte des avis clients  sur des plateformes spécialisées telles que :

- TrustedShops
- TrustPilot
- Amazon

Nous avons considéré les critères suivants pour sélectionner notre jeu de données :

- Volumétrie des données disponibles
- Volumétrie des réponses du fournisseur

# Etape 0 - Collecte des données

## Source des données:

Afin de collecter les données sur les avis clients, nous avons exploré 3 plateformes permettant de récupérer des avis clients avec différents outils ou scripts python :

- Amazon
- Trustpilot
- TrustedShop

Notre choix final s'est tourné vers la **plateforme TrustedShops** pour laquelle nous avons de la flexibilité pour extraire des données des avis clients sur les entreprises ciblées sans changer notre code de web scraping.

Pour des considérations de volumétrie des avis clients et de volumétrie des réponses des fournisseurs, nous avons décidé de réaliser l'étude avec les données d'avis clients sur La Redoute, leader français du e-commerce en mode et maison.

## Méthodes explorées:

Nous avons testé plusieurs méthodes/outils de web scraping pour collecter les avis clients:

- Réalisation de scripts python avec BeautifulSoup
- Usage d' extensions de navigateur, par exemple le data instance scraper de google
- Applications spécialisées de web scraping telles que Scrapfly qui permettent de bypasser les captcha ou les anti bot.

Nous avons réalisé notre propre programme de web scraping en python avec Beautiful, afin de mettre en pratique les notions de web scraping vus en BootCamp.

Échantillons d'extractions disponibles :

- Rayon d'Or Bagage : https://www.rayondor-bagages.fr/ : 11 colonnes x 1082 lignes
- Mechatechnic : https://www.mecatechnic.com/fr-FR/ : 11 colonnes x 6245 lignes
- La Redoute: https://www.trustedshops.fr/evaluation/info_XA3F5BBDE34B5AD3FB64015EB823AC7C6.html : 11 col x 34760 lignes

## Web Scraping avec BeautifulSoup:

Le programme de web scraping réalisé se déroule en deux grandes étapes :

- **Étape 1** : **Scraper les données brutes des avis clients sur TrustedShops pour l'entreprise ciblée et les stocker dans un fichier csv.**

  Cette étape est réalisée par la fonction *get_reviews(url, n_pages, output_file)* avec :

  - *url* : url de la 1ere page de résultats sur les avis clients pour l'entreprise ciblée
  - *n_pages* : nombre de pages de résultats que l'on veut récupérer. Ce nombre doit être inférieur au nombre total de pages web de résultats disponibles pour l'entreprise ciblée.
  - *output_file* : nom du fichier csv dans lequel sont stockés les les avis clients collectés.

- **Étape 2 : Réaliser un premier nettoyage et formatage des données**

  Cette 2eme est réalisée à l'aide de la fonction *clean_reviews(input_file, output_file)* avec :

  - *input_file* : fichier csv de données brut collecté dans l'étape 1
  - *output_file* : fichier csv final nettoyé et formaté

  La fonction clean_review réalise principalement les actions suivantes :

  - Suppression des colonnes inutiles : "provider", "inModeration", "verificationStatus"
  - Conversion des dates timestamps au format date
  - Extraction des réponses des fournisseurs aux avis clients

Le détail du programme de web scraping des avis TrustedShops est disponible dans le notebook "01_mdp_trustedshops_scraping_v1.0.ipynb" sur GitHub.

# Étape 1 - Exploration des données et Data Visualisation

## Exploration des Données :

Voici un aperçu de ce que nous avons extrait pour le site de la Redoute, en termes de données brutes. Environ 35000 ratings sont disponibles pour environ 27400 commentaires.

```
<class 'pandas.core.frame.DataFrame'>
Index: 34760 entries, rev-cc907a30-4fb0-4e55-a4a1-307b89
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   rating             34760 non-null  int64
 1   title              27666 non-null  object
 2   comment            27399 non-null  object
 3   createdAt          34760 non-null  object
 4   transaction        30737 non-null  object
 5   updatedAt          32 non-null     float64
 6   totalLikeCount     8 non-null      float64
 7   customerCity       3189 non-null   object
 8   SupplierReply      6801 non-null   object
 9   SupplierReplyDate  6801 non-null   object
dtypes: float64(2), int64(1), object(7)
memory usage: 2.9+ MB
```

## Préparation des données:

Après extraction des informations, nous avons effectué **un premier nettoyage** de données en supprimant les valeurs manquantes et les colonnes non pertinentes ou les pourcentages de Nan sont trop importants.

Nous avons aussi enlevé les lignes sans commentaires, les lignes ou les dates de transaction étaient manquantes.

Cf ci-dessous :

```
display(val_manquantes)
```

| | nb_valeurs_manquantes | pct_val_manquantes |
|---|---|---|
| **title** | 7094 | 0.204085 |
| **comment** | 7361 | 0.211766 |
| **transaction** | 4023 | 0.115736 |
| **updatedAt** | 34728 | 0.999079 |
| **totalLikeCount** | 34752 | 0.999770 |
| **customerCity** | 31571 | 0.908257 |
| **SupplierReply** | 27959 | 0.804344 |
| **SupplierReplyDate** | 27959 | 0.804344 |

## Enrichissement du jeu de données :

Ensuite, nous avons souhaité **approfondir la dimension temporelle e**n extrayant les mois, jours ouvrés/week-end ainsi que les heures auxquelles les commentaires ont été postés.

- date_of_cmt
- year_of_cmt
- month_of_cmt
- cmt_in_weekend
- hour_of_cmt
- year_of_transaction
- month_of_transaction
- day_of_transaction
- weekday_of_transaction
- transaction_in_weekend
- transaction_date

Nous avons aussi ajouté les variables suivantes qui nous permettent de calculer le nombre de mots des commentaires du texte ou du titre:

- comment_length
- log_comment_length
- title_length

Le log est une fonction logarithmique appliquée aux commentaires pour étudier la distribution des données en fonction du rating.

## Data Visualisation

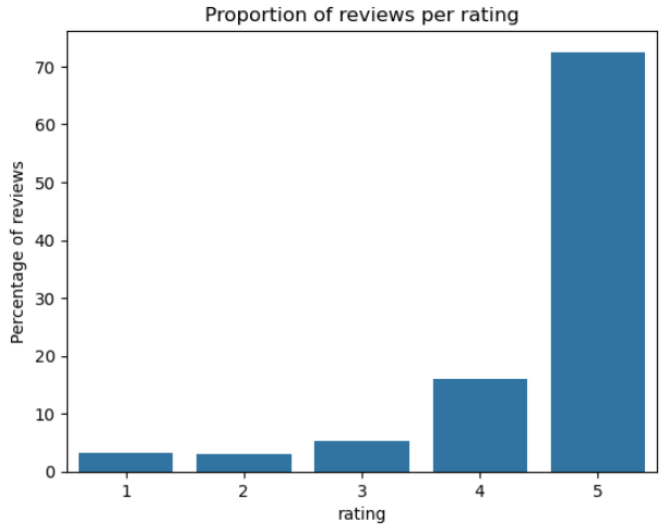
Nous avons ensuite grâce aux outils de visualisation connus cherché à comprendre comprendre la distribution de ces variables, leurs corrélations et plus particulièrement la relation entre le rating et les autres variables.

L'ensemble des graphiques se retrouve ici mais dans le cadre du compte-rendu écrit, nous avons décidé de choisir les plus pertinents.

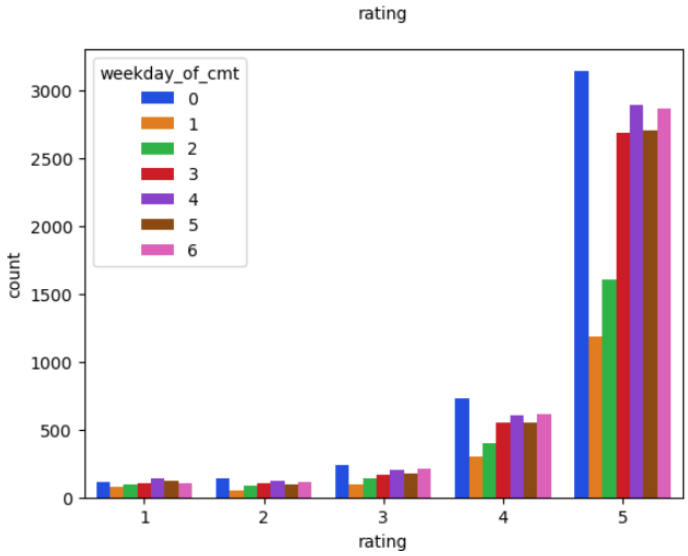
### Analyse de la distribution des variables:

#### Graphique 1 : Proportion de reviews par rating:

On constate ici, avec surprise, que la **majorité des commentaires se fait lorsque le rating est extrêmement positif, càd à 5.** En d'autres termes, lorsque les clients sont satisfaits, ils sont plus enclins à commenter.



Graphique 2: Nb de commentaires en fonction du jour de la semaine :

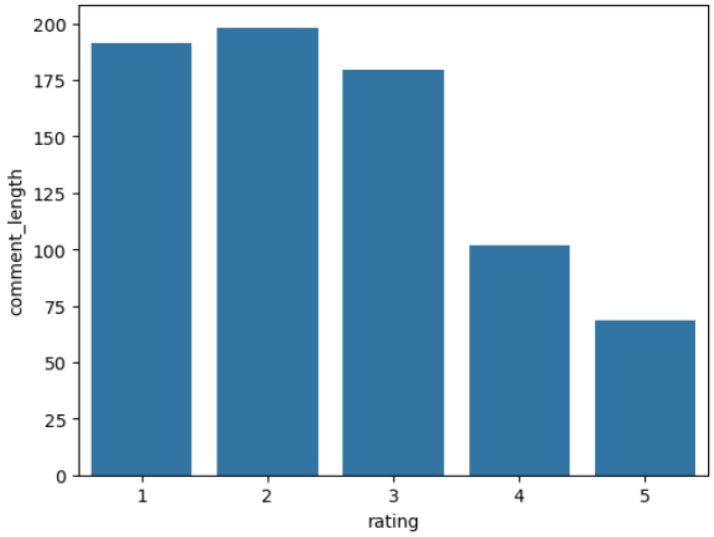



Complétés avec d'autres graphiques non retranscrits ici, on constate que :

- Les revues ont principalement été effectuées en 2023
- La distribution des ratings par mois est presque uniforme pour les notes allant de 1 à 3. Par contre, il y a plus de disparités sur les notes de 4 et 5.
- Il y a des **pics de commentaires en mars, avril, juillet, septembre et octobre** pour la plupart des ratings mais surtout pour ceux allant de 4 à 5.
- **La plupart des revues sont faites entre le vendredi et le lundi**
- Les revues sont principalement faites dans l'après-midi, entre midi et 14h, sinon le matin entre 6h et 10h

Graphique 3: Longueur moyenne du commentaire =f(rating)

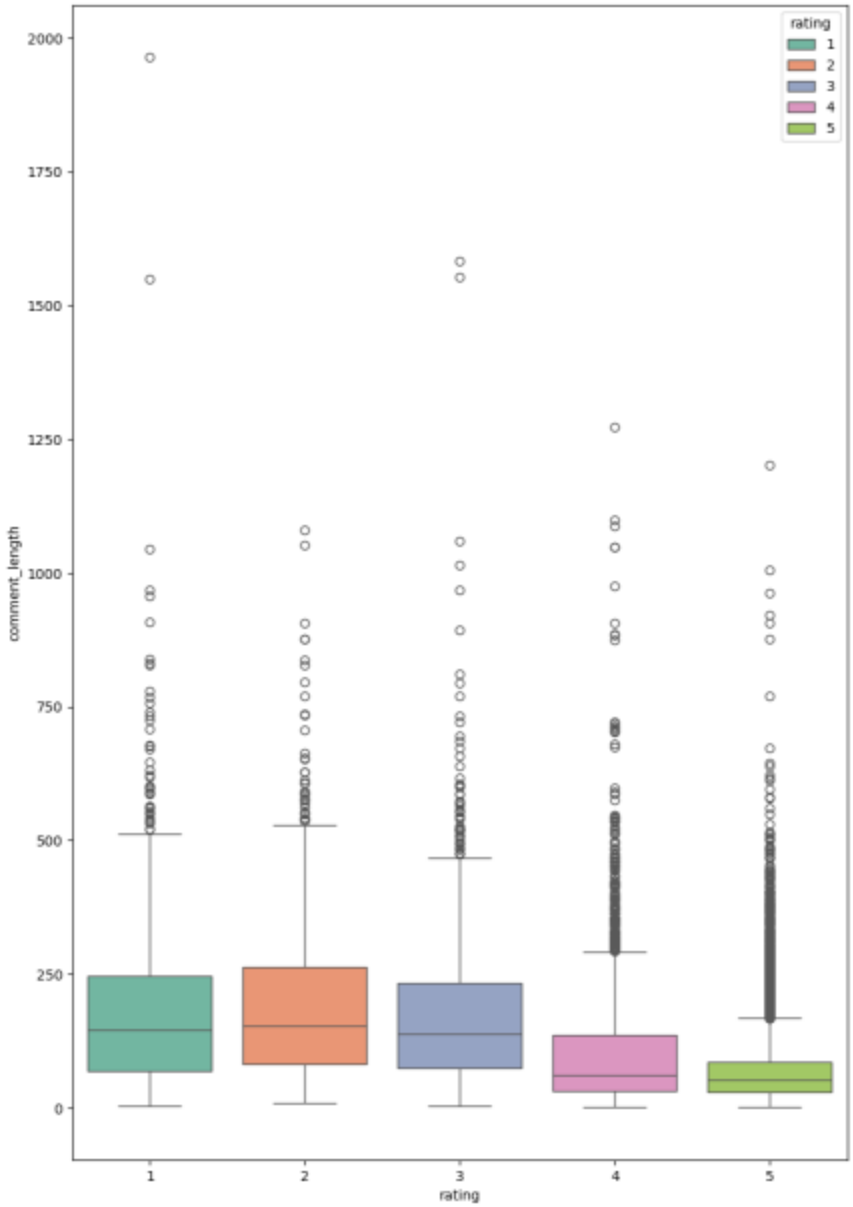
| rating | comment_length |
|---|---|
| 1 | 191.348505 |
| 2 | 198.260083 |
| 3 | 179.239643 |
| 4 | 101.514392 |
| 5 | 68.750439 |




En moyenne, **on constate que les revues avec des notes inférieures ou égales à 3 ont tendance à avoir des commentaires plus longs**, ce qui paraît logique car le client manifeste son insatisfaction et utilise plus de mots pour la décrire.

**La moyenne de la longueur du commentaire diminue lorsque la note s'améliore.**

Graphique 4: Boîte à moustaches pour vérifier les outliers et les médianes:




Avec ce type de graphique, on constate que **la médiane de la longueur de commentaires est nettement plus basse pour les notes de 4 et de 5** mais aussi qu'il y a une longue queue d'outliers avec asymétrie positive.

Graphique 5 : Word cloud sur les  commentaires

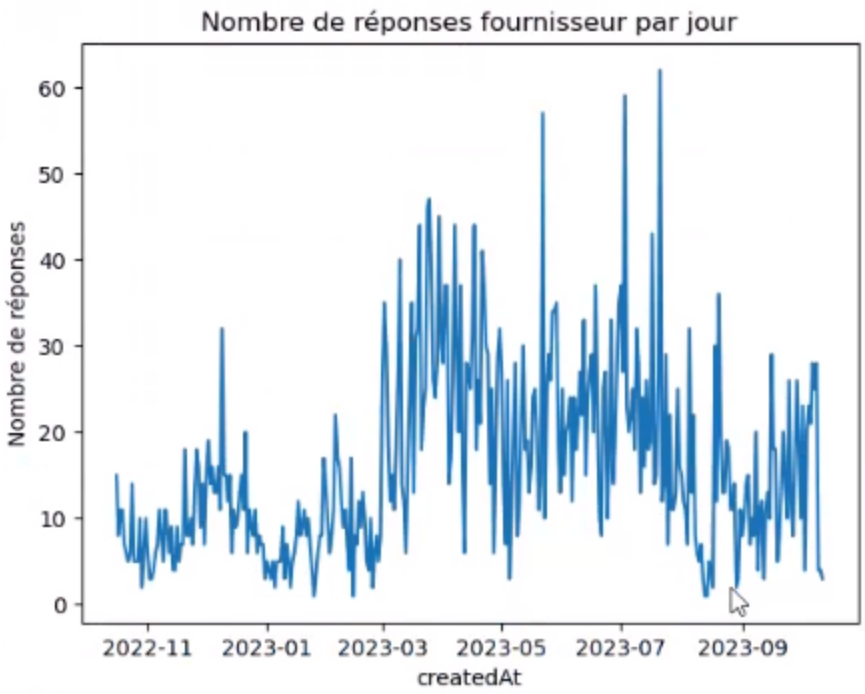
On constate qu'il y a d**u nettoyage de texte à faire pour avoir un rendu plus significatif**.

Il est aussi important de noter qu'un **mot pris hors de son contexte ne nous aide pas à prédire le sentiment du client.**

Par exemple, le mot "livraison" est mentionné très fréquemment mais on ne sait pas conclure si la livraison s'est bien passée ou si finalement elle a eu du retard ou si le colis a été endommagé.

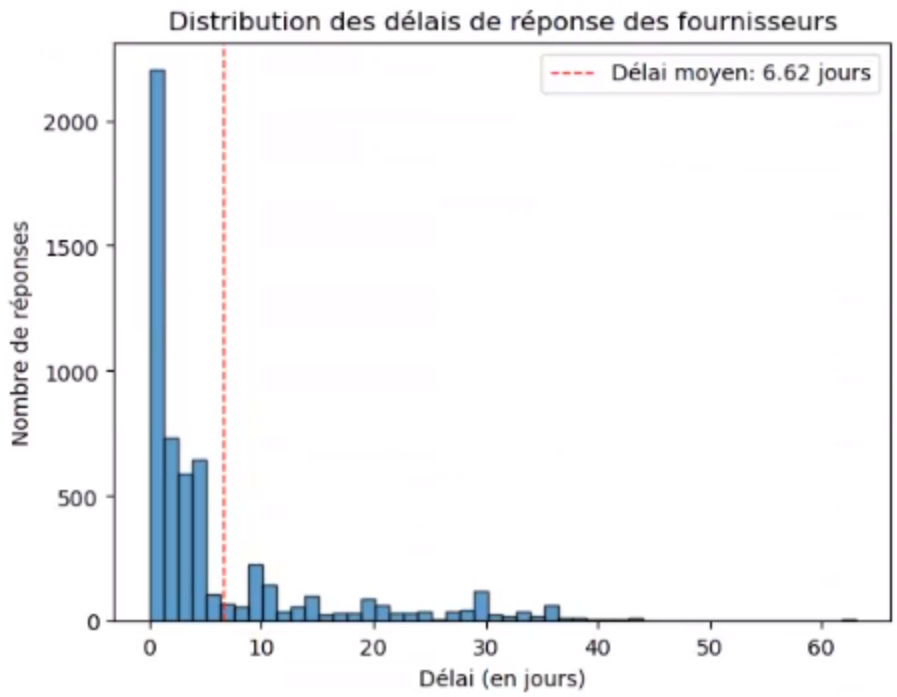
## Analyse des Réponses fournisseurs :

Graphique 6 : Nombre de réponses fournisseur par jour :



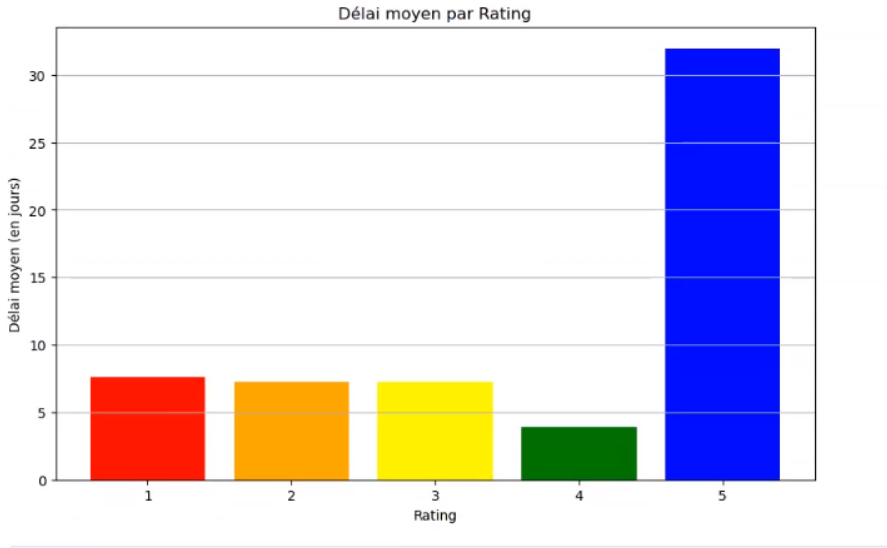

On constate une augmentation des réponses entre mars et juillet 2023, pour ensuite les voir redescendre sur les périodes suivantes.

Graphique 7: Distribution des délais de réponse des fournisseurs :




On constate que la grande majorité des réponses se fait en dessous du délai moyen de 6.62 jours avec un étalement de ces dernières pouvant aller au-delà des 60 jours.

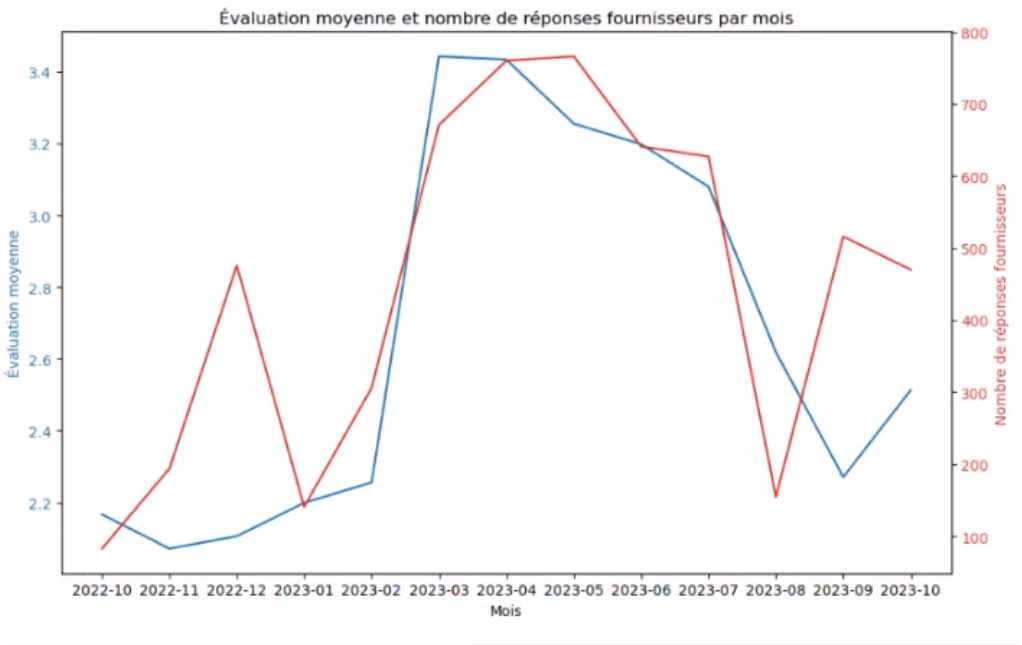
Graphique 8: Analyse du délai moyen par Rating:



L'analyse du délai moyen des réponses nous montre une rupture pour la note de 5 ou le délai de réponse devient très long. Cela peut s'expliquer par le fait qu'une bonne appréciation du client n'impacte pas la réputation du fournisseur et est par conséquent traitée en seconde priorité.

En ce qui concerne les notes allant de 1 à 3, nous sommes presque constants en termes de délai de réponse.

Graphique 9: Evaluation moyenne et nombre de réponses fournisseurs par mois:

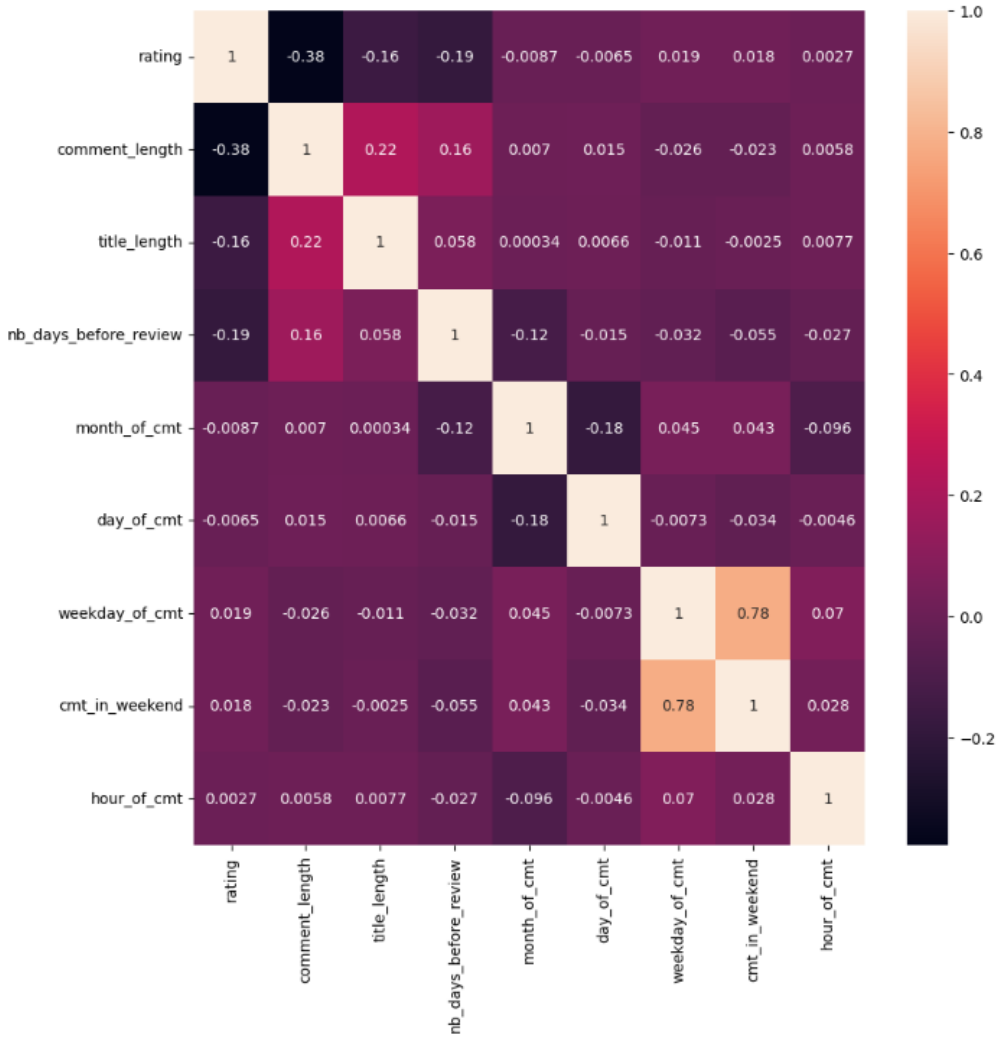



Ce dernier graphique confirme que le nombre de réponses fournisseurs suit la même tendance que l'évaluation moyenne des clients, en tout cas à partir de février 2023. Avant cette date, il y a moins de données disponibles, ce qui peut expliquer les divergences entre les deux courbes.

## Analyse de corrélation :

L'objectif étant d'identifier les variables corrélées à **notre variable cible : le rating.**

Graphique 10 : Analyse de corrélation par heatmap :




Ici, on constate les **premières corrélations entre la longueur du commentaire ('comment_length') et le rating, avec une valeur de -0.38.**

Il y aussi 2 autres corrélations moins marquées avec le rating et qui sont pour les variables : **'nb_days_before_review' et 'title_length**', respectivement avec les valeurs -0.19 et -0.16.

## Tests statistiques:

Notre premier objectif  est de valider statistiquement la relation entre :

- le rating et les commentaires,
- Le rating et les dates de revue,
- Le rating et la longueur du titre

Le deuxième objectif est de vérifier la nature de la corrélation entre les variables explicatives.

Nous avons réalisé plusieurs tests pour identifier des corrélations entre les différentes variables.

- Test de Anova : nous n'avons pu appliquer Anova qu'entre la fonction logarithmique de la longueur du commentaire et  le rating qui suivent une distribution normale confirmée par un diagramme QQ plot.

Pour les autres variables comme la longueur du titre et le nombre de jours avant revue, nous n'avons pas pu l'utiliser. A ce stade,  **nous pouvons confirmer que  le rating a un effet significatif sur la longueur du commentaire.**

- Test de Kruska Willis : ce test nous permet de confirmer  scientifiquement la relation statistique **entre le rating et la longueur des commentaires mais aussi entre les titres et le nombre de jours entre la transaction et la date du commentaire.**

- Test de Spearman : le test de Spearman nous fait conclure qu'**il y a une corrélation entre la longueur du commentaire et le rating, mais aussi entre la longueur du titre du commentaire et le nombre de jours entre la transaction et le commentaire ( nb_days_before_review).**

A ce stade, nous avons donc **trois variables explicatives possibles  pour prédire le rating** :

- La longueur du commentaire client
- La longueur du titre du commentaire
- Le nombre de jours entre la date du commentaire et la date de la transaction.

Afin de confirmer qu'il n 'y a pas de redondance entre ces variables explicatives candidates, nous avons analysé la colinéarité entre ces 3 variables à l'aide du VIF (Variance Inflation

Factor qui est une mesure du degré de colinéarité entre les variables explicatives dans un modèle de régression multiple).

*Calcul du VIF avec variance_inflation_factor de statsmodels.stats.outliers_influence*.

| | VIF | variable |
|---|---|---|
| 0 | 5.464626 | Intercept |
| 1 | 1.003351 | title_length |
| 2 | 1.003351 | nb_days_before_review |

| | VIF | variable |
|---|---|---|
| 0 | 3.031557 | Intercept |
| 1 | 1.027085 | comment_length |
| 2 | 1.027085 | nb_days_before_review |

**Le VIF calculé vaut pratiquement 1 pour chacune des variables. Donc, nos trois variables ne sont pas colinéaires.**

## Synthèse de l'exploration des données et de leur visualisation

La première partie de nos analyses nous fait constater les choses suivantes :

- **Constat 1 - Analyse des ratings :** les notes excellentes (ratings 4 et 5) sont sur-représentées. On risque donc d'avoir un biais sur cette information car 70% des commentaires se font lorsque la note est excellente.

Il faudra prendre en compte ce déséquilibre des données dans la phase de modélisation et sûrement réfléchir à un ré-échantillonnage des données.

- **Constat 2 - Analyse des commentaires clients :** l'analyse de mots hors contexte ne nous permet pas de conclure sur la satisfaction du client et il faudra par conséquent utiliser des outils plus sophistiqués capables de regrouper les mots ( word embedding?)

  Il va falloir réfléchir à comment encoder du texte en numérique pour être capable d'utiliser les outils de NLP.

- **Constat 3 - Relation entre les variables et le rating :** L'analyse exploratoire et les tests statistiques nous ont permis de valider les points suivants :
  - La longueur du commentaire, la longueur du titre du commentaire, le nombre de jours entre la transaction et l'émission de l'avis sont statistiquement corrélés au rating
  - Le calcul du VIF montre que la longueur du commentaire, la longueur du titre du commentaire, le nombre de jours entre la transaction et l'émission de l'avis ne sont pas colinéaires.

Par conséquent, nous pourrons choisir ces 3 variables comme variables explicatives pour prédire le rating.

- **Constat 4 - Analyse des réponses fournisseurs :**
    - On constate que les réponses semblent être génériques. Ce qui confirme l'intérêt de générer des réponses plus adaptées et même de prédire quels seraient les templates à utiliser.
    - L'évaluation moyenne des fournisseurs et le nombre de réponses fournisseurs par mois varient mensuellement, avec un pic autour de mars 2023. A confirmer lorsque nous aborderons les notions de détection de saisonnalité.
    - Le délai moyen de réponse s'allonge avec l'amélioration de l'évaluation et une note de 5 peut conduire à des délais extrêmement longs.
    - La majorité des délais de réponse se situent en dessous de 10 jours, avec un délai moyen de 6,62 jours.
    - Le nombre de réponses fournisseurs par jour présente une tendance à la hausse jusqu'en mai 2023, suivie d'une légère baisse.

  Ces informations **suggèrent une corrélation entre le nombre de réponses et l'évaluation des fournisseurs. Elles indiquent aussi des variations dans les délais de réponse selon l'évaluation et également une corrélation entre le nombre de réponses et la saisonnalité.**

# Etape 2 - Pre Processing des données

Nous avons réalisé certaines étapes d'enrichissement de notre jeu de données lors de la phase d'analyse.

Pour rappel, nous avons **exploré la dimension temporelle** afin d'isoler les jours, mois, années, semaines, week-end et heures des commentaires.

Aussi, nous avons créé de nouvelles variables pour **mesurer la longueur** - en nombre de mots - **du titre et des commentaires.**

Pour le moment nous n'avons pas croisé ce premier jeu de données de TrustedShops avec d'autres sources d'information.

# Etape 3 - Modélisation

Dans cette étape de modélisation, nous nous sommes concentrés principalement sur trois objectifs: :

- **Prédire la note du client** en fonction de ses commentaires
- **Prédire la réponse du fournisseur** en fonction de la note du client
- **Réaliser une analyse de sentiments** des commentaires clients et fournisseurs

## Prédiction de la note du client en fonction des commentaires

### Problématique

1. Analyse de sentiment et Problématique de classification binaire: prédire si la note est positive ( notes 4 et 5) ou négative (notes 1, 2, 3).

   La variable cible ici est le sentiment avec:

$$sentiment = 0 \ si\, rating < 4$$

$$sentiment = 1 \ si\, rating > 3$$

2. Analyse de sentiment et problématique de classification multi-classes : prédire si la note est positive (notes 4 et 5), neutre (note 3) ou négative notes (1 et 2)

   La variable cible ici est :

$$sentiment = -1 \ si\, rating < 3$$

$$sentiment = 0 \ si\, rating = 3$$

$$sentiment = 1 \ si\, rating > 3$$

Dans la suite on utilisera le terme sentiment négatif :

- pour les ratings < 4 dans le cas de classification binaire.
- Pour les ratings < 3 dans le cas de classification multi - classes.

Pour rappel, l'échantillon de données est très déséquilibré avec les ratings 4 et 5 représentant plus de 80% de l'échantillon.

## Démarche

Nous avons suivi une approche graduelle pour modéliser les données. Nous avons commencé par une première modélisation utilisant les 3 features identifiées lors de l'analyse exploratoire. Ensuite, nous avons appliqué des modèles NLP de "Bag of Words" de vectorisation de texte pour plus de performance. Enfin, nous avons utilisé des modèles de réseaux de neurones récurrents pour améliorer la qualité de la prédiction.

## Métriques de performance utilisées

Afin d'évaluer nos modèles et de comparer leurs performances, nous avons utilisé les métriques suivantes:

- Accuracy, Precision, Recall et f1_score.
- Recall et Précision des modèles pour la classe des notes négatives : ratings 1, 2 ou 1, 2, 3 en fonction de la problématique binaire ou multiclasses. Le but étant de pouvoir capter correctement et prédire avec précision le sentiment négatif.
- Balanced_Accuracy,
- Recall macro, f1_macro , Précision-macro, Recall weighted, f1_weighted, Précision Weighted pour pouvoir apprécier la performance des modèles sur l'ensemble des classes de rating tout en intégrant le déséquilibre des données,
- Les temps d'entraînement et de prédiction des modèles,
- Les matrices de confusion.

<u>Afin de départager les modèles testés, nous avons pris en compte 5 métriques:</u>

- **Accuracy** : la proportion de prédictions correctes par rapport à toutes les prédictions,
- Le **Recall** pour la classe du sentiment négatif (recall_0), une mesure de la proportion de sentiments négatifs réels qui sont correctement identifiés,
- **Précision** pour la classe du sentiment négatif (precision_0) qui reporte la proportion de sentiments négatifs prédits qui sont réellement corrects,
- **Recall - macro,**
- **f1-macro**

Nous cherchons des modèles avec une bonne accuracy et ayant une capacité à capturer les sentiments négatifs tout en ayant une précision performante dans la prédiction du sentiment négatif.

## Modèles de référence dit Baseline models:

La variable cible est le sentiment définit plus haut.

Les variables explicatives sont :

- Longueur du titre du commentaires,
- Longueur du commentaire,
- Nombre de jours entre la date de transaction et la date de création du commentaire.

### Modèles testés

Nous avons testé une dizaine de modèles et comparé leurs performances :

- Régression Logistique
- Arbre de décision
- Forêt aléatoire
- Gaussian Naïve Bayes (Gaussian NB)
- Multinomial Naïve Bayes (Multinomial NB)
- Complement Naïve Bayes (Complement NB)
- Linear SVC
- SVC
- KNeighbors Classifier
- Gradient Boosting

### Standardisation des données

Les données d'entraînement et de test ont été standardisées. Pour tous les modèles en dehors du Multinomial NB, nous avons centré et réduit les données. Pour le modèle Multinomial NB nous avons appliqué une transformation Min Max.

### Sous-échantillonage aléatoire

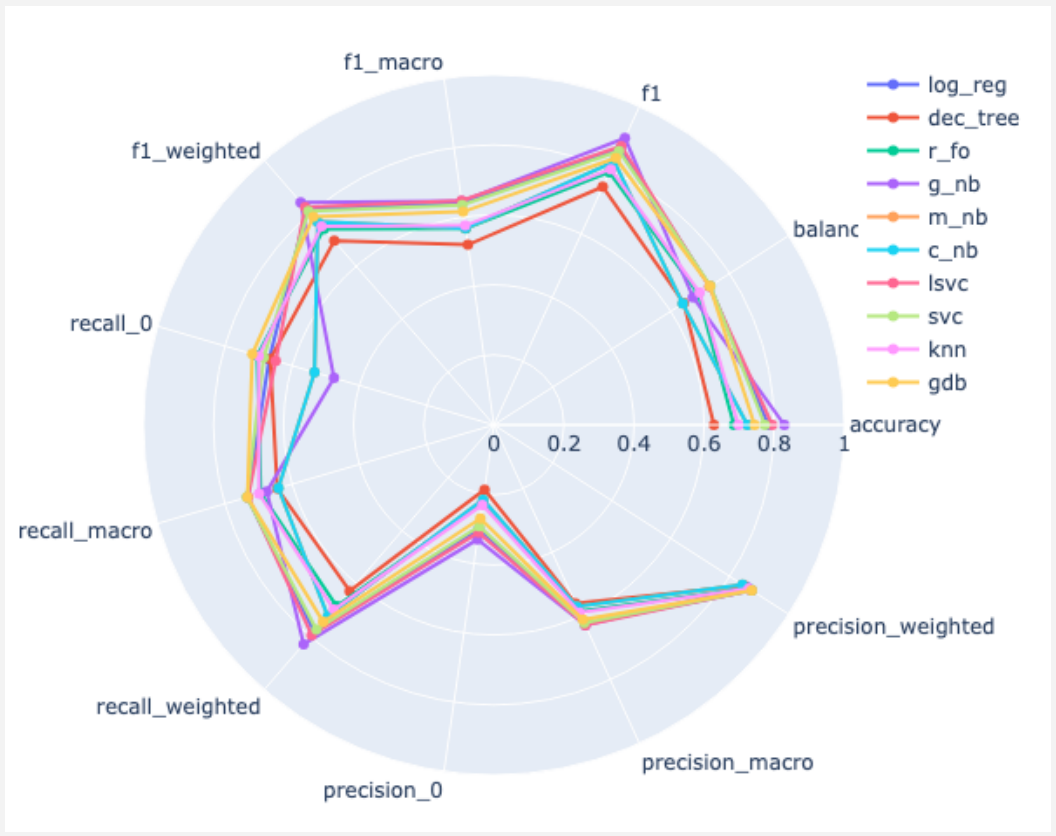
Pour tenir compte du déséquilibre des classes nous avons appliqué un sous-échantillonnage aléatoire des données d'entrainements et de test.

### Choix du modèle et Optimisation

- Recherche des meilleurs hyperparamètres et validation croisée avec GridSearchCV pour obtenir des modèles plus performants et plus stables en termes de généralisation.
- Usage de pipeline pour étanchéifier la chaîne de traitements entre les données d'entraînement et les données de test.

## Résultats pour la classification binaire

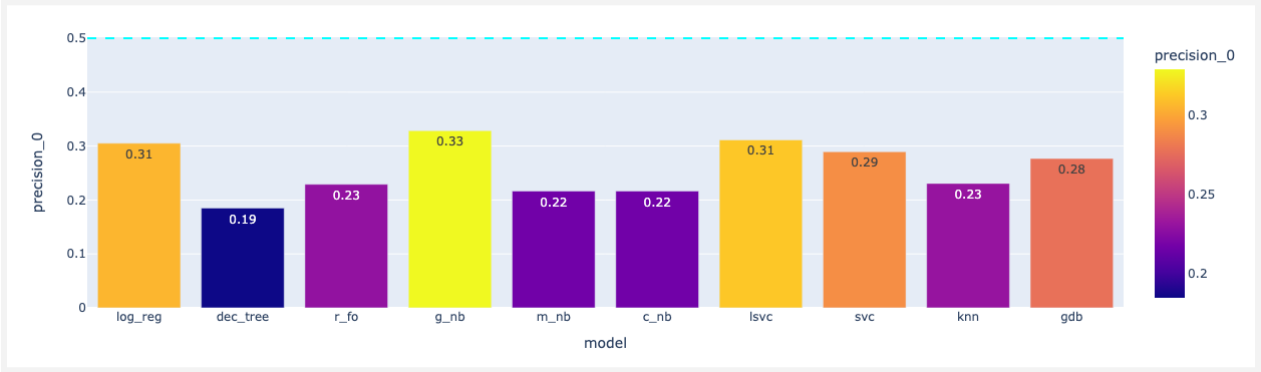
Graphique 11 - Radar de comparaison des métriques




**Les 10 modèles testés ont une précision d'ensemble (accuracy) supérieure à 60% et avoisinant les 80% pour les plus performants.**

Le recall de la classe de sentiment négatif (recall_0) est supérieur à 60% pour la majorité des modèles , hormis pour Complement NB et Gaussian NB . Donc ces modèles sont capables de capter correctement au moins 60% des sentiments négatifs.

Graphique 12 - Précision dans la prédiction du sentiment négatif



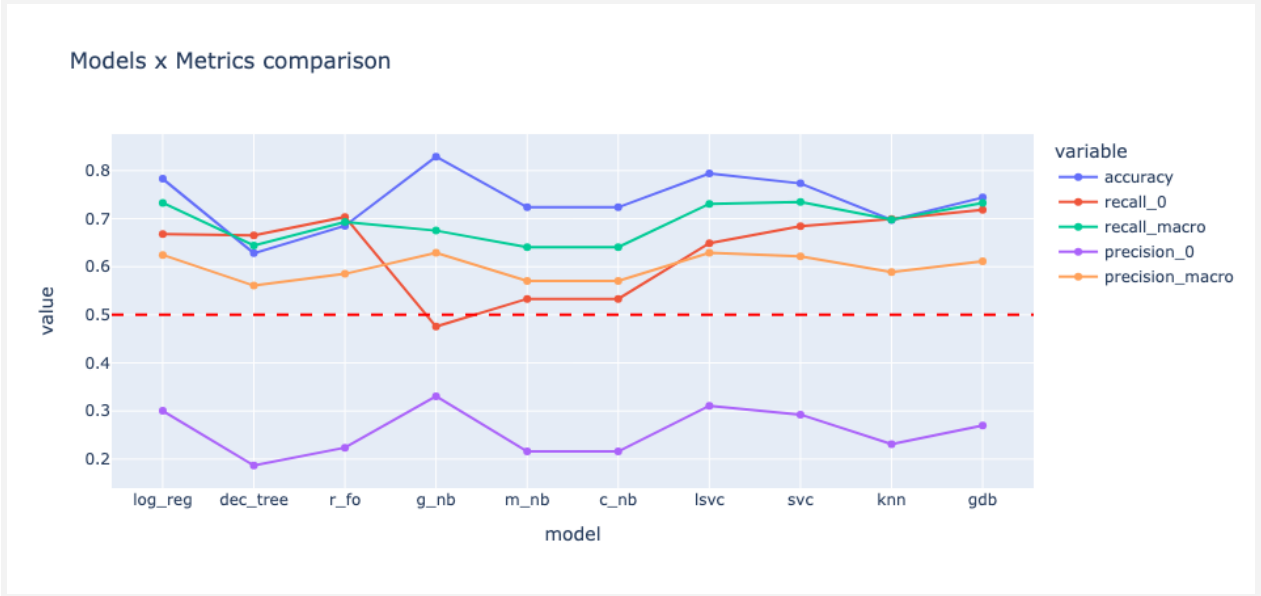

**Par contre, la précision pour la classe de sentiment négatif est en dessous de 33%, i.e. Au plus un tiers des sentiments prédits comme négatifs sont réellement négatifs.**

Graphique 13 - Temps de d'entraînement vs Temps de prédiction

Les modèles les plus coûteux en temps de prédictions sont SVC, KNN, Forêt Aléatoire.

Les modèles les plus coûteux en temps d'entraînement sont SVC, Forêt aléatoire(r_fo) et Gradient Boost (gdb).

Graphique 14 - Comparaison sur 5 métriques

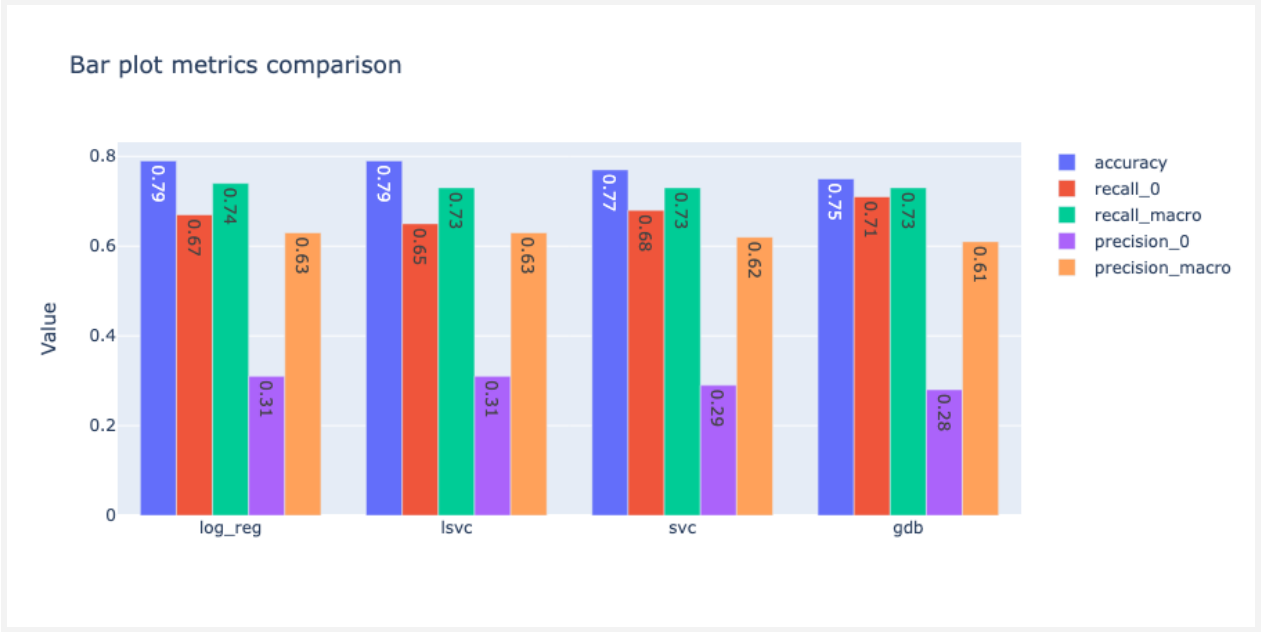



Les modèles les plus performants en terme de précision d'accuracy de recall_0, de precision_0 , de recall macro et précision macro sont :

- Régression logistique (log_reg)
- Machine à Support de vecteur SVC
- LinearSCV (lsvc)

- Gradient Boost (gdb)

Graphique 15 - métriques top 4




Dans nos 4 modèles les plus performants, le gradient boosting (gdb) et SVC ont un meilleurs recall_0 (72% et 68% respectivements) par contre les modèles Régression Logistique(log_reg) et LinearSVC (lsvc) ont une meilleure precision_0 à 31%.

<u>Conclusion sur la modèle de base de classification binaire</u>

**Prenant en considération le temps d'entraînement, le temps de prédiction et la simplicité, nous retenons le modèle de régression logistique.**

Le modèle de régression logistique nous permet de prédire si un commentaire est positif ou négatif en se basant sur des éléments simples comme la longueur du titre, la longueur du texte du commentaire et le temps pris par le client pour donner un retour. **Ce modèle a une précision globale de 79%, ce qui signifie qu'il prédit correctement le sentiment dans 79% des cas.**

Il est capable de capter jusqu'à 67/ des sentiments négatifs. En revanche, sa précision dans la prédiction du sentiment négatif est assez faible. Seul un tiers des sentiments qu'ils prédit comme étant négatifs sont réellement des sentiments négatifs.

## Modèles “Bag of Words” avec Tf-Idf ET CountVectorize

Le texte des commentaires est transformé en données numériques avec les techniques NLP Tf-IdF et CountVectorizer. Ces données numériques sont utilisées comme variables explicatives pour prédire le sentiment avec les modèles de classifications classiques.

**CountVectorizer** est une technique qui compte simplement le nombre d’occurrences de chaque mot dans un document. Il **crée un vecteur pour chaque document où chaque élément du vecteur représente le nombre d’occurrences d’un mot particulier** dans le document. Cette technique est simple et rapide, mais elle ne prend pas en compte l’importance relative des mots dans le document.

**TF-IDF (Term Frequency-Inverse Document Frequency)** est une technique qui prend en compte l’importance relative des mots dans le document. Elle **calcule un score pour chaque mot qui est proportionnel à la fréquence du mot** dans le document et inversement proportionnel à la fréquence du mot dans l’ensemble des documents. Les mots qui apparaissent fréquemment dans un document mais rarement dans l’ensemble des documents ont un score plus élevé, ce qui signifie qu’ils sont plus importants pour la compréhension du document.

### Démarche

**1 - Traitement de texte**

Pour Appliquer le modèle “Bag of Words” nous appliquons des traitements NLP au texte des commentaires en 4 étapes :

- Étape 1 - Nettoyage des commentaires à l’aide des regex python: Il s’agit d’enlever les caractères spéciaux, la ponctuation, les chiffres, les mots de moins de 3 lettres.
- Etape 2 - Exclusion des stop-words avec NLTK : Il s’agit d’expurger des commentaires les mots qui n’apportent pas d’information pertinente pour notre problématique (e.g. les articles comme le, la , les , ...)
- Étape 3 - Lemmatization avec SpaCy: Il s’agit de remplacer chaque mot par son lemme qui correspond à sa signification racine. Cette étape permet de réduire le nombre total de mots.
- Etape 4 - Racinisation (Stemming) avec NLTK : Il s’agit de réduire chaque mot à sa forme de base.

**2 - Transformation du texte en vecteur numérique avec Count Vectorize et TfidfVectorizer**

La vectorisation avec Count Vectorize et Tf-IdF est alors appliquée au texte des commentaires où chaque mot a été réduit à sa racine. Nous obtenons donc une matrice V où les colonnes correspondent aux mots de l’ensemble du corpus des commentaires.

Chaque ligne de la matrice correspond à un commentaire représente :

- La fréquence de chaque mot du commentaire dans le corpus de l'ensemble des commentaires dans le cas de Countvectorizer
- Au score TF-IDF de chaque mot commentaire dans le corpus de l'ensemble des commentaires. Le score est élevé pour les mots qui sont fréquents dans un commentaire particulier, mais rares dans l'ensemble du corpus des commentaires. Cela aide à donner plus de poids aux mots qui sont probablement plus significatifs pour le document, tout en réduisant l'importance des mots communs qui apparaissent dans de nombreux documents, comme "le", "la", "et", etc.) dans le cas d'une Vectorization avec Tf-IdF.

**V représente donc les variables explicatives et le sentiment défini dans la problématique plus haut est la cible à prédire.**

**3 - Séparation données d'entraînement et données de test.**

Lorsque nous séparons les données en données d'entraînement et données de test, nous prenons le soin de nous assurer que la répartition des classes de sentiment est conservée entre les données de test et les données d'entraînement. **Utilisation de l'option stratify dans le fonction train test split de sklearn.**

**4 - Sous - échantillonnage aléatoire**

Les données étant déséquilibrées, nous appliquons un sous-échantillonnage aléatoire comme dans le modèle de base pour prendre en compte le déséquilibre des données.

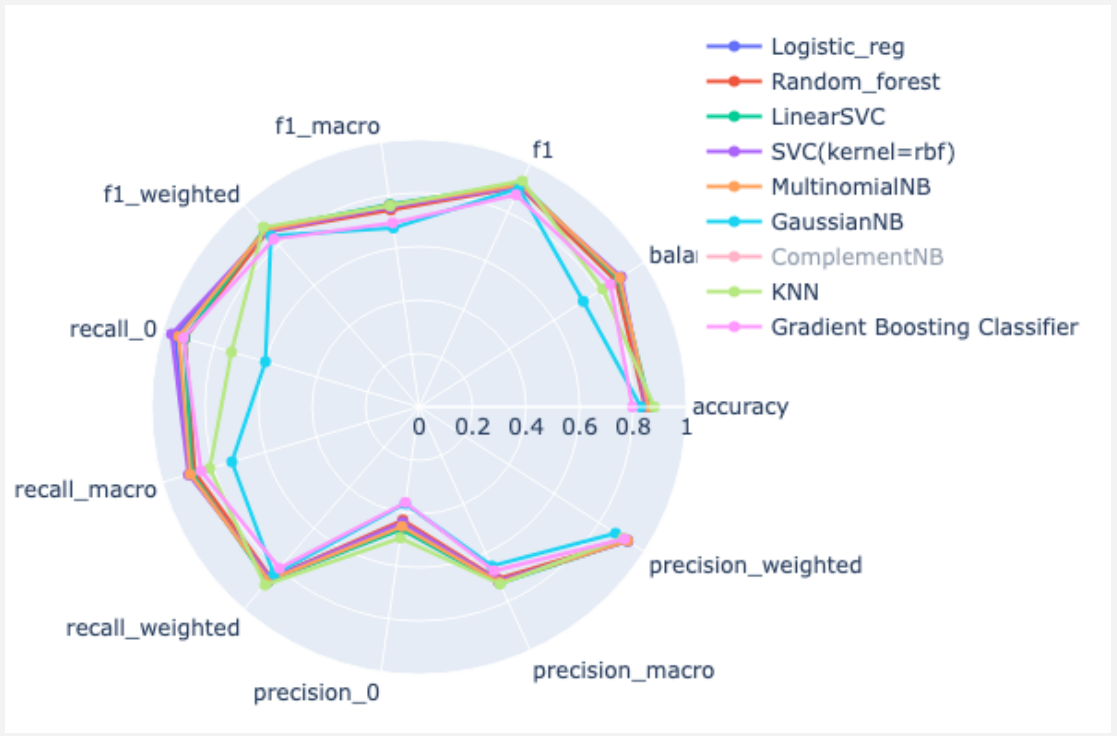
**5 - Application des 10 modèles de classification** utilisés dans la modélisation de base et comparaison des résultats et sélections des 3 modèles les plus performants.

**5 - Tentative d'optimisation avec Grid Search CV et utilisation de pipeline** (TF IDF Vectorize -> Random Undersampling - > Model ) pour une sélection de modèles.

**6 - Tentative d'optimisation avec un réseau dense de neurones.** Afin d'obtenir une meilleure précision pour le sentiment négatif, nous appliquons un modèle de réseaux de neurones denses aux commentaires vectorisés avec Tf-IdF.

## Résultats pour TF-IDF Vectorize

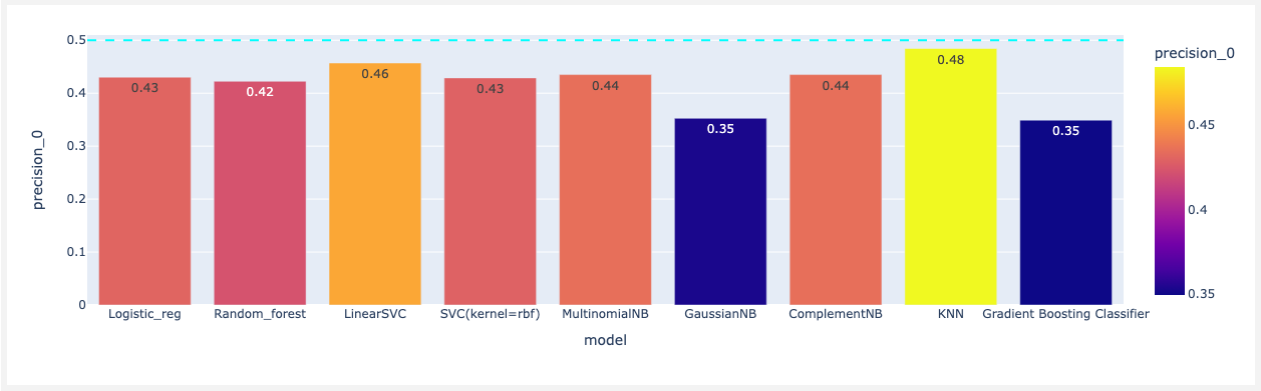
Graphique 16 - radar des métriques des modèles




**On constate une nette amélioration des performances dans l'ensemble par rapport au modèle de base. La précision globale de l'ensemble des modèles est supérieure à 78%.**

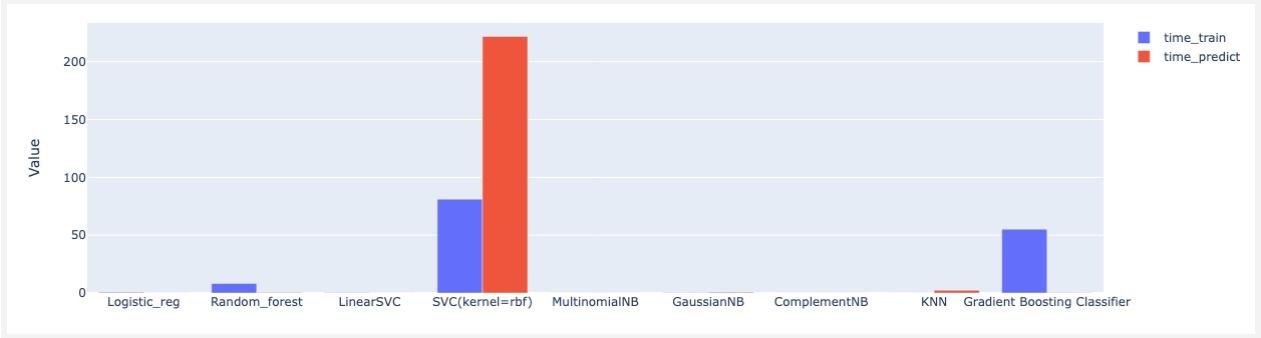
Quasiment tous les modèles , en dehors du Gaussian Naïve Bayes, sont capables de détecter au moins 74% des sentiments négatifs.

Graphique 17 - Précision pour la classe des sentiments négatifs



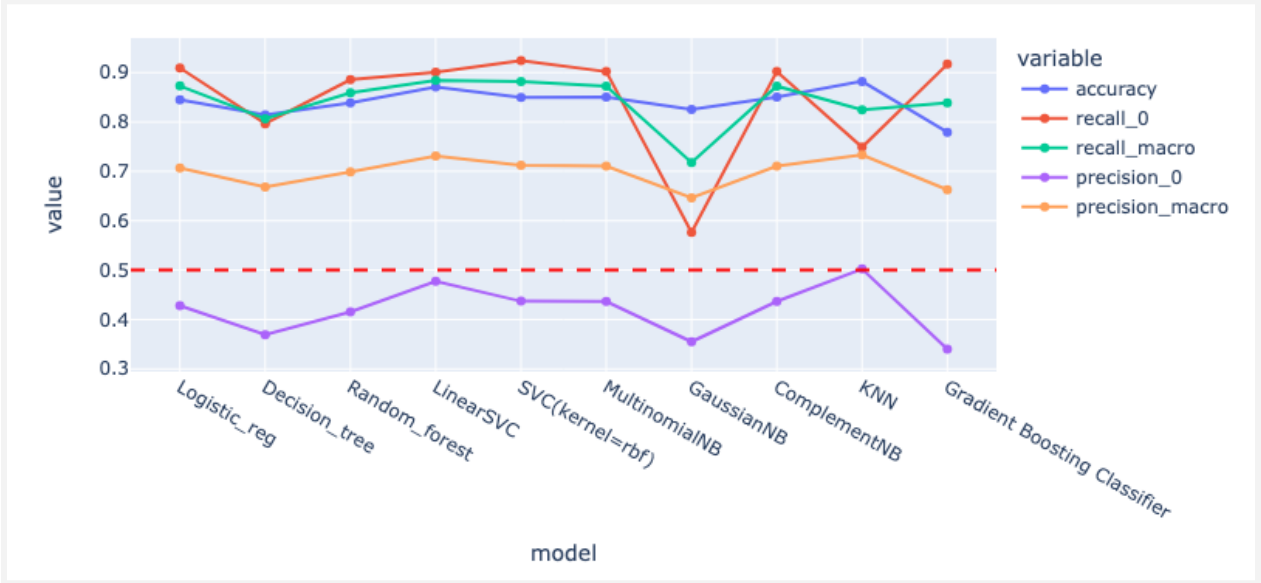

La précision dans la prédiction du sentiment négatif reste faible même s'il y a une nette amélioration par rapport au modèle de base avec des valeurs entre 35% et 48%.

Graphique 18 - temps d'entraînement vs temps de prédiction




Le SVC reste le modèle le plus lent à entraîner et à prédire

Graphique 18 - Comparaison pour 5 métriques



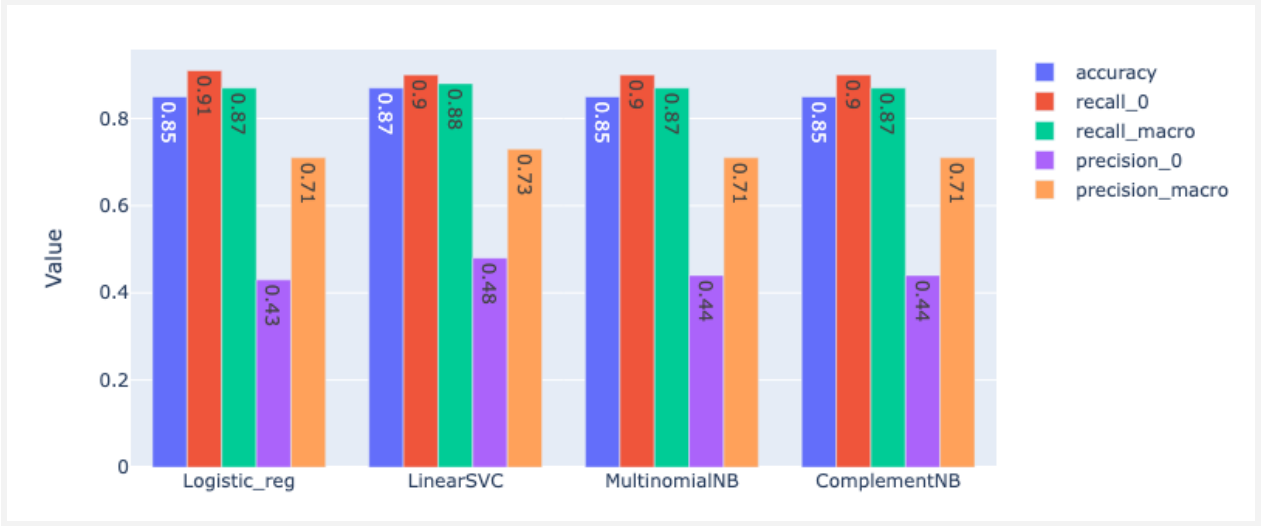

**Les 4 modèles les plus performants par rapport au 5 métriques accuracy, recall_0, recall_macro, precision_0 et precision_macro et en prenant en compte le temps de prédiction sont :**

- **Logistic Regression**
- **LinearSVC**
- **Multinomial Naïve Bayes**
- **Complement Naïve Bayes**

Notons que le modèle KNN a certes la meilleure precision_0, mais il a le plus faible recall_0 à 75% contre plus de 90% pour les modèles sélectionnés.

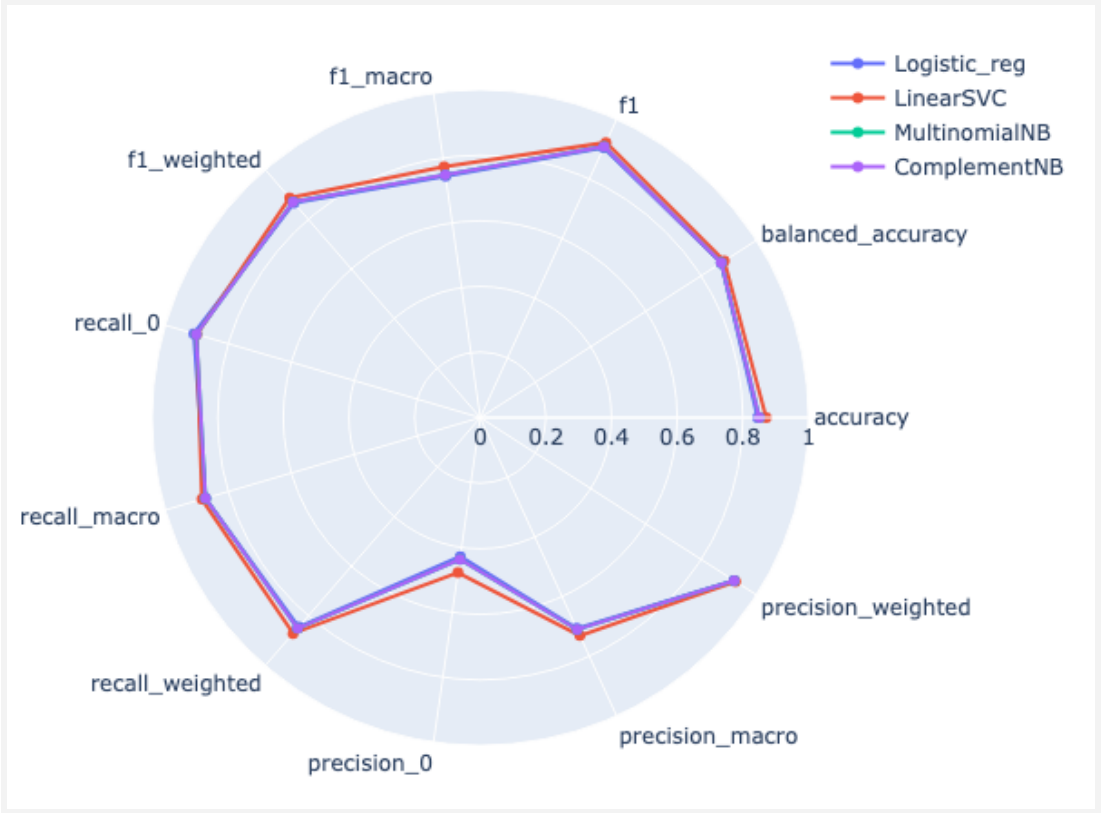
Le graphique suivant permet de visualiser plus clairement les performances des 4 modèles.

Graphique 19 - Métriques du top 4



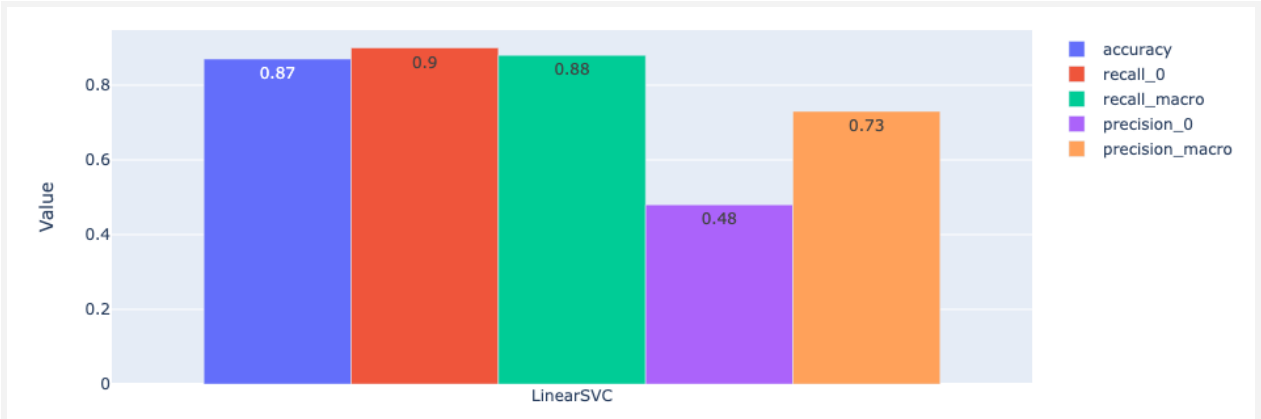

Le Graphique ci-après permet de départager clairement les 4 modèles sélectionnés:

Graphique 20 - Radar des métriques du top4




**Le modèle LinearSVC se démarque nettement des 3 autres modèles avec une meilleure accuracy, une meilleure precision_0, un meilleur f1_macro**.

Graphique 21 - Métriques du modèle LinearSVC

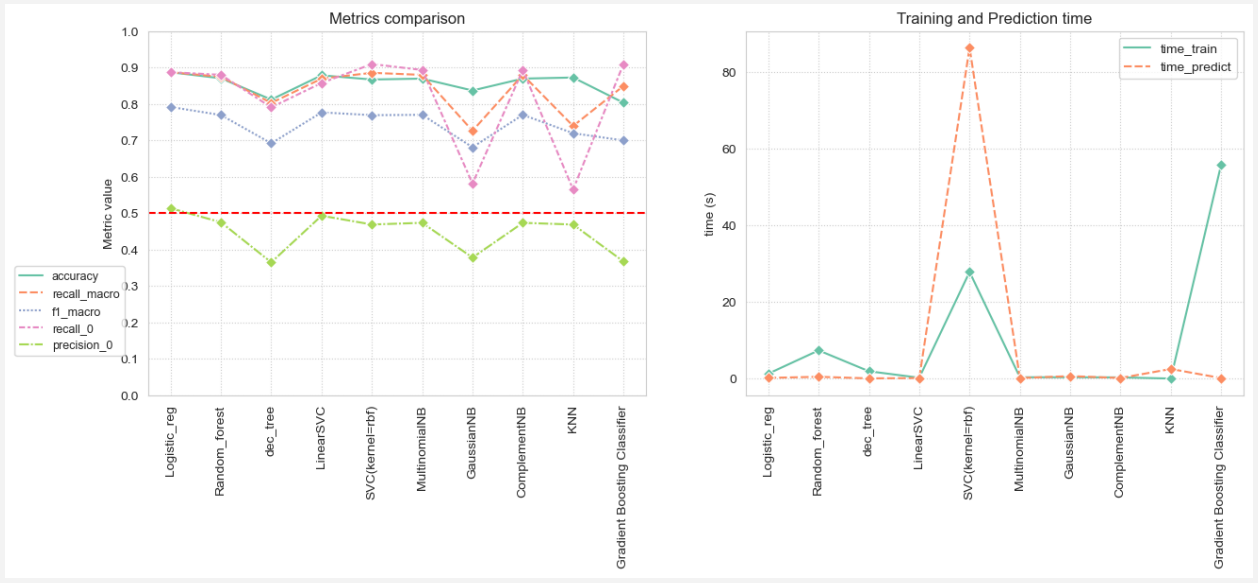


Conclusion sur le Bag of Words avec Tf-IdF

**Nous constatons une amélioration très nette des performances avec le Bag Words et Tf-IdF Vectorize comparé au modèle de base.** Le modèle LinearSVC sélectionné réussi à prédire correctement 87% des sentiments en général. De plus, il est capable de capter 90 % des sentiments négatifs. Cependant, il y a encore une marge d'amélioration car 52% des sentiments qu'il identifie comme négatifs sont en réalité positifs.

## Résultats pour Count Vectorize

Graphique 22 - Comparaison des métriques et temps



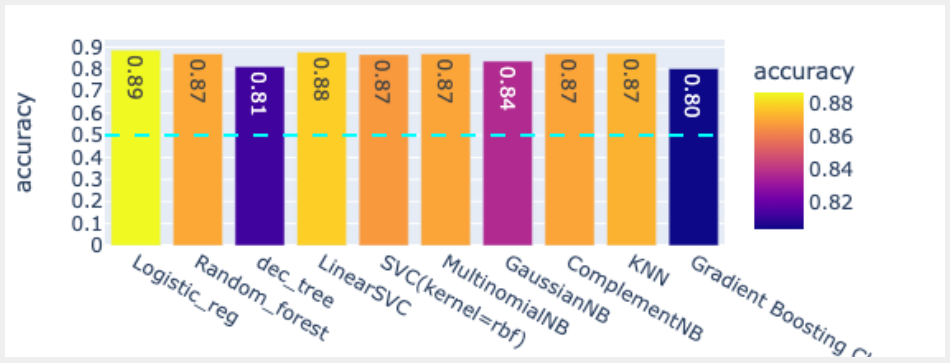

Dans l'ensemble, l'accuracy et la précision pour la classe sentiment négatif est meilleure qu'avec TfidfVectorizer. Le modèle de régression logistique atteint une précision de 51% pour le sentiment négatif.

La précision d'ensemble est aussi meilleure et se rapproche des 90% pour le modèle le plus performant, la régression logistique.

Par contre, avec Count Vectorizer on capture un petit peu moins bien les sentiments négatifs qu'avec le Tf idf Vectorizer.

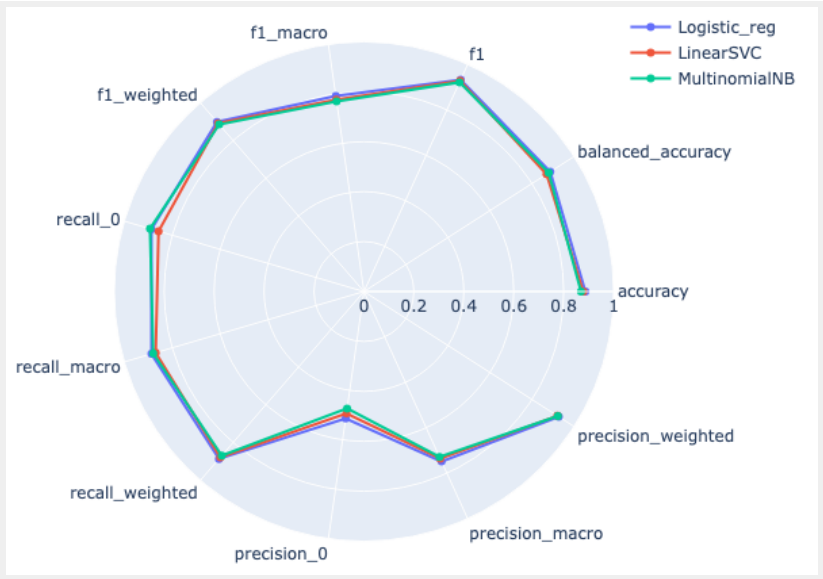
**Si on tient compte du temps d'entraînement et de prédiction, les 3 meilleurs modèles sont : Régression Logistique, LinearSVC et Multinomial NB**.

Graphique 24 - Précision globale des modèles



Graphique 25 - Métrique du top 3

Graphique 25 - radar top 3 des modèles




Il ressort visuellement que le modèle de régression logistique se démarque des deux autres en termes de performance.

Conclusion sur le Bag of Words avec CountVectorizer

**Nous constatons encore une amélioration très nette des performances avec le Bag Words et CountVectorizer comparé au modèle de base.** La capacité à prédire le sentiment dans l'ensemble et la précision dans la prédiction du sentiment négatif est meilleure qu'avec TfidfVectorizer. Néanmoins, Tfidf Vectorizer a une meilleure capacité à détecter les sentiments négatifs. Ses performances restent très proches des performances de Tf-IdF.

Le modèle de régression Logistique avec Count Vectorizer  réussit à prédire correctement 89% des sentiments en général. De plus, quand il prédit un sentiment négatif, il a raison 51% du temps. Cependant, il détecte un petit peu moins bien les sentiment négatifs qu'avec Tf idf Vectorizer.

## Résultats de l'optimisation avec Grid Search CV

Nous appliquons un GridSearchCV aux modèles de Régression Logistique, LinearSVC et MultinomialNB pour tenter d'obtenir de meilleures performances. Ces trois modèles ont de bonnes performances et ne sont pas très longs à entraîner.

Nous incluons aussi l'étape de vectorization TfidfVectorizer dans le GridSearchCV afin d'élargir l'espace de recherche des hyper-paramètres.

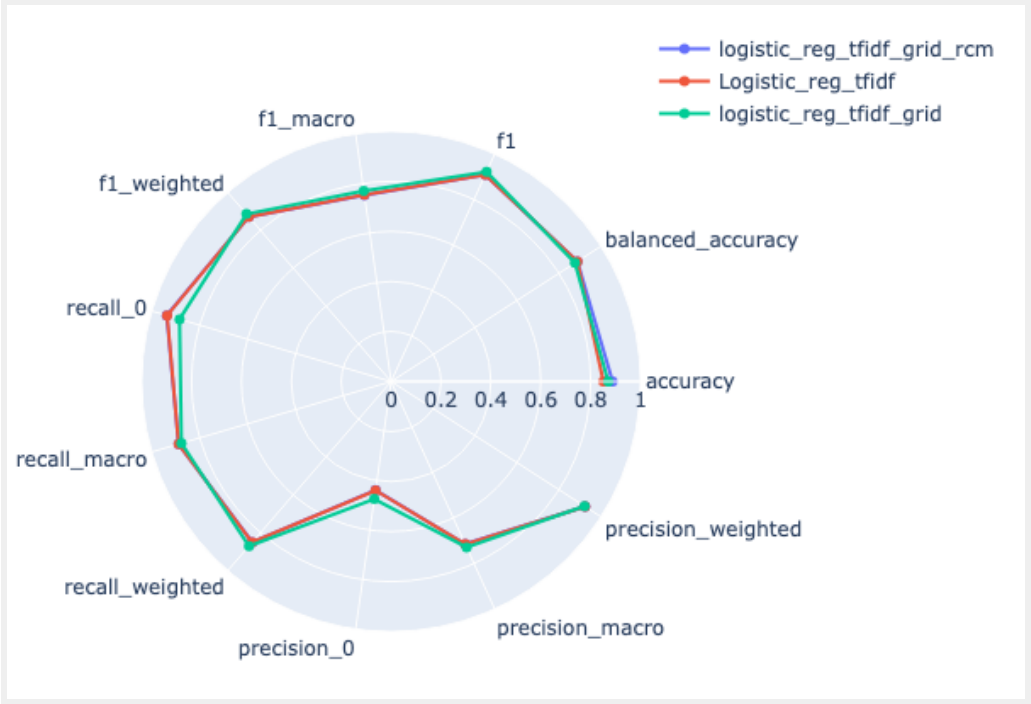
Pour ce faire, pour chaque modèle nous définissons des pipelines à l'aide de la fonction make_pipeline de la librairie imblearn :

- *pipe_lsvc = make_pipeline(TfidfVectorizer(), RandomUnderSampler(), svm.LinearSVC())*
- *pipe_lreg = make_pipeline(TfidfVectorizer(), RandomUnderSampler(), LogisticRegression())*
- *pipe_mnnb = make_pipeline(TfidfVectorizer(), RandomUnderSampler(), MultinomialNB())*

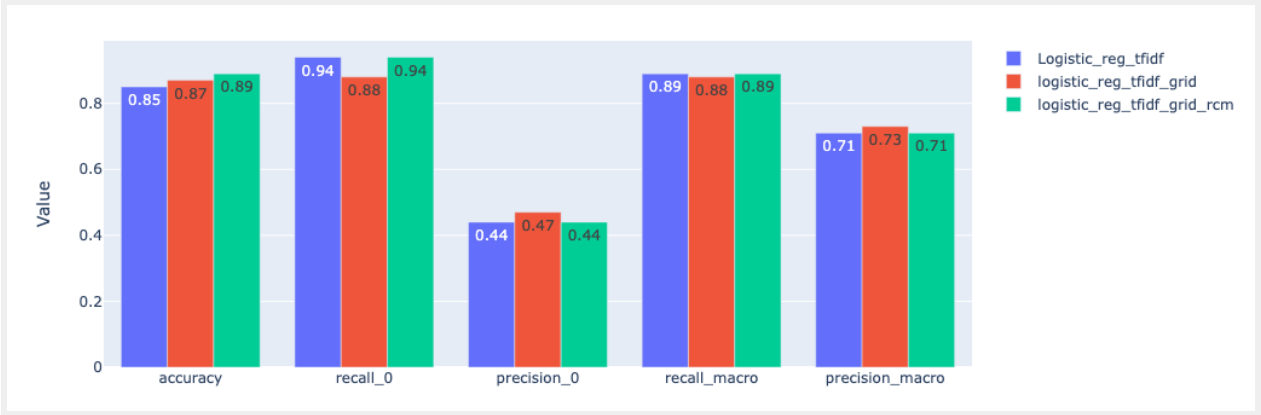
Nous realison deux GridSearchCV

- Un recherche avec condition de l'accuracy (grid)
- Une recherche avec condition du recall_macro (grid_rcm)

Graphique 26 - Radar - Comparaison avec GridSearchCV




Graphique 27 - Barplot - Comparaison avec GridSeachCV

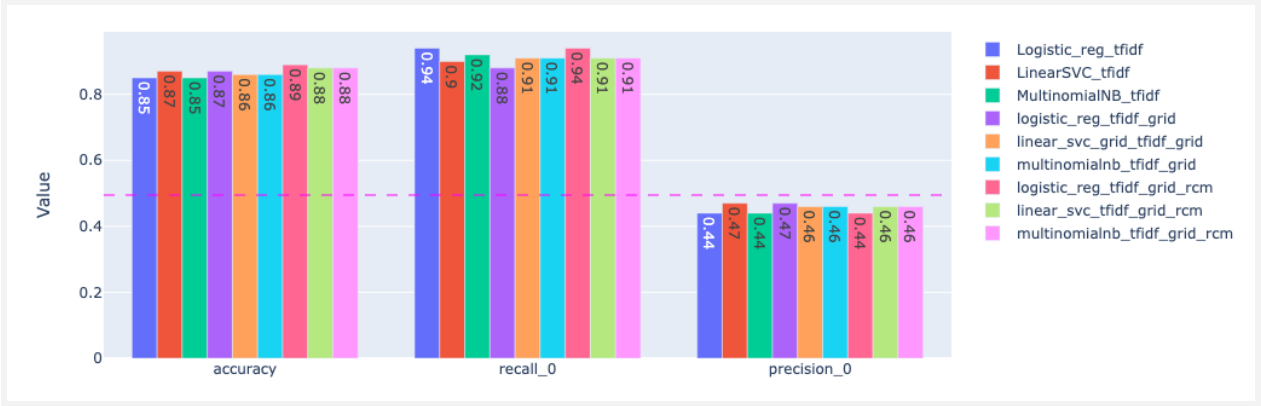



Le GridSearchCV avec optimisation sur l'accuracy apporte une amélioration de la précision d'ensemble, une meilleure précision dans la prédiction du sentiment négatif, mais régresse dans la capacité à détecter les sentiments négatifs.

Le GridSearchCV avec optimisation sur le recall macro apporte une amélioration de la précision sans changements sur les autres métriques de performance.

**Dans l'ensemble il n'y a pas d'amélioration notable des performances.** Comme on peut constater dans le graphique ci-après, la précision dans la prédiction des sentiments négatifs reste toujours en deçà de 50% et la capacité à détecter le sentiment négatif ne change pas vraiment.

Graphique 28 - Group Barplot - Comparaisons avec sans GridSearchCV




## Application d'un réseau dense de neurones

Après une transformation avec TfidfVectorizer, nous appliquons un réseau dense de neurones à 2 couches :

$$model = models.Sequential()$$

$$model.add(layers.Dense(32,\ activation = 'relu',\ input_shape = (V_ru.shape[1],)))$$

$$model.add(layers.Dense(32,\ activation = 'relu'))$$

$$model.add(layers.Dense(1,\ activation = 'sigmoid'))$$

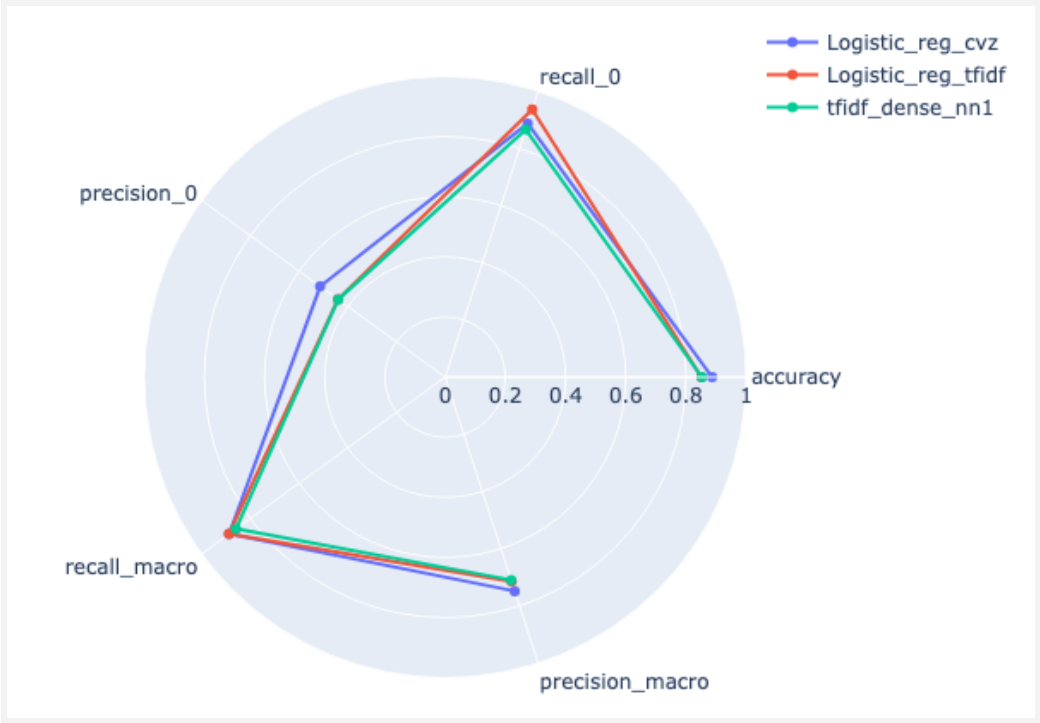
$$Fonction\ de\ perte:\ 'binary_crossentropy'$$

$$Optimiseur:\ Adam$$

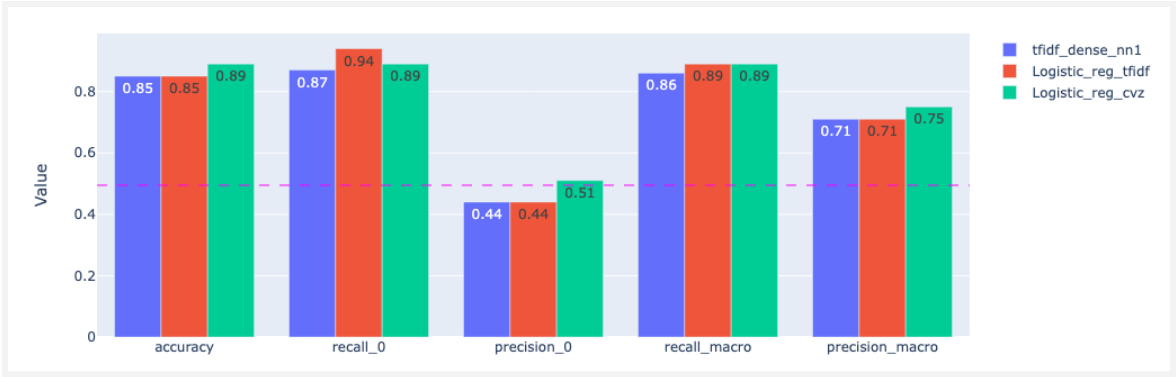
$$Métrique:\ accuracy$$

## Résultats du modèle Tf Idf & Réseau dense de neurones

Graphique 29 - Réseau de Neurone Dense vs Régression Logistique




Graphique 30 - Barplot - Réseau de neurones dense vs Régression Logistique




**Bien que les performances du réseau de neurones testé soient dans les mêmes ordres de grandeur que celles de la régression logistique, la régression logistique avec Counter Vectorizer ou TfidfVectorizer offre de meilleures performances.**

## Modèle de réseau de neurones - GRU : Gated Recurrent Units

Afin d'améliorer la qualité de nos prédictions, notamment la précision, nous nous sommes tournés vers les réseaux de neurones récurrents.

### Qu'est ce qu'un Réseau de Neurones Récurrent?

"Un Réseau de Neurones Récurrent (RNN) est un modèle d'apprentissage automatique puissant qui permet d'analyser des séquences de données, telles que du texte, de la parole ou des séries temporelles. Ces réseaux permettent aux machines de « se souvenir » des informations passées et de les utiliser pour prendre des décisions en temps réel." cf. DataScientest.

### Qu'est ce qu'un GRU?

Les GRU sont une variante des réseaux de neurones récurrents qui permettent de capturer les dépendances à long terme dans les séquences de données.

Un GRU est comme un lecteur intelligent qui lit un texte mot par mot. Pour chaque nouveau mot, il décide quels détails du passé sont importants à retenir et quels détails moins pertinents il peut oublier. Il fait cela grâce à un système de "portes" qui s'ouvrent et se ferment pour contrôler le flux d'informations.

C'est cette capacité à gérer efficacement la mémoire à court et long terme qui rend les GRUs plus performants que les RNNs traditionnels pour analyser des séquences de données plus longues, comme le texte ou la parole.

### Démarche de mise en oeuvre d'un modèle GRU

**Etape 1. Nettoyage du texte des commentaires**.

Pour le modèle de GRU mis en œuvre, nous avons nettoyé le texte des commentaires en enlevant les caractères spéciaux, la ponctuation, les stop Word. Nous avons aussi enlevé les mots rares, i.e. qui apparaissent moins de deux fois dans l'ensemble des commentaires.

**Etape 2. Tokenization et numérisation**

Il s'agit de découper l'ensemble des commentaires nettoyés en mots. On obtient un corpus de mots de l'ensemble des commentaires nettoyés.

Chaque mot est ensuite converti en un nombre. Ainsi chaque commentaire devient une séquence de nombres.

**Etape 3. Padding (pad sequence)**

Les commentaires sont désormais des séquences de nombres. Ces séquences ont des longueurs différentes. Le padding consiste à normaliser la longueur de ces séquences en une longueur définie. Nous avons choisi la taille de la plus longue séquence pour notre modèle.

À l'issue nous obtenons une matrice X_pad dont chaque ligne est la séquence à longueur normalisée des commentaires. X_pad est notre variable explicative pour tenter de prédire le sentiment avec un réseau de neurones Gated Recurrent Unit.

**Etape 4. Définition du modèle GRU.**

Nous avons testé plusieurs modèles. Nous donnons ici un exemple illustratif , le modèle qui a obtenu la meilleure précision dans la prédiction du sentiment négatif.

Graphique 31 - Modèle GRU_1

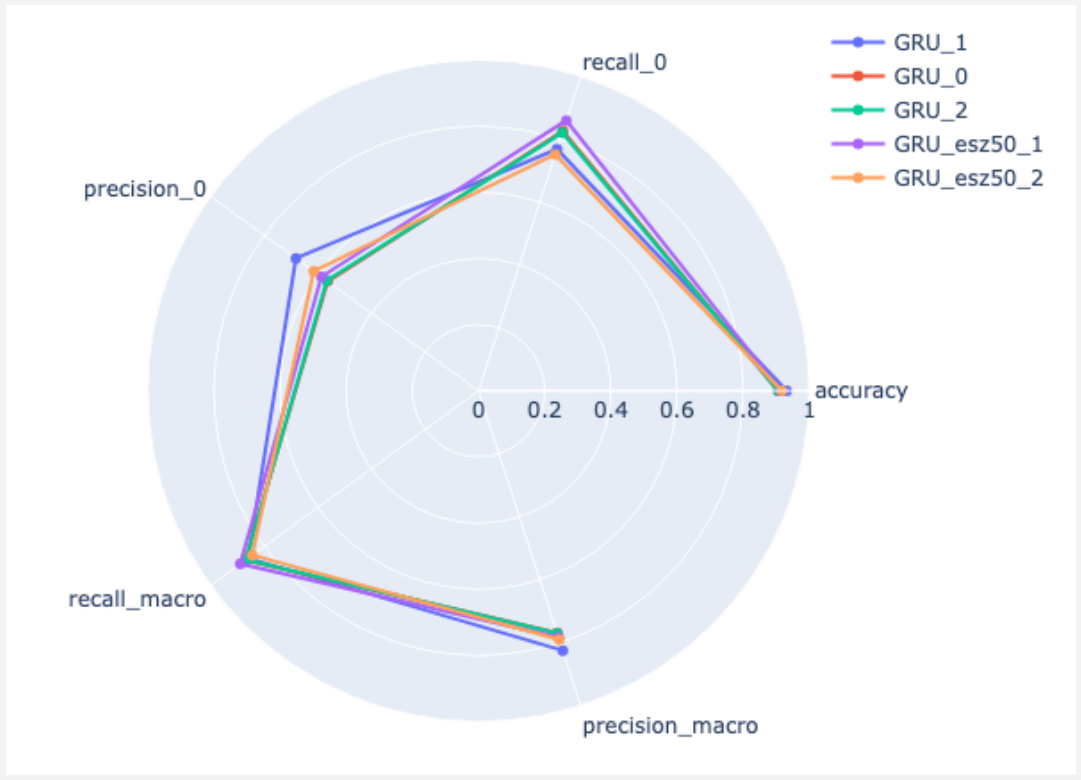
```
model_name = "GRU_1"
embedding_size = 100
model2 = Sequential()

model2.add(Embedding(input_dim=num_words,
                     output_dim=embedding_size,
                     input_length=max_tokens,
                     name="embedding_layer"))

model2.add(GRU(units=128, return_sequences=True))
model2.add(GRU(units=64, return_sequences=True))
model2.add(GRU(units=12))
model2.add(Dense(1, activation='sigmoid'))

# Compile
optimizer = Adam(learning_rate=0.008)
model2.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
```

**Pour la prise en compte du déséquilibre des données, nous avons utilisé le paramètre *class_weight*** *( paramètre dans la fonction fit de Keras utilisé pour pondérer l'importance de chaque échantillon en fonction de la classe à laquelle ils appartiennent, pendant*

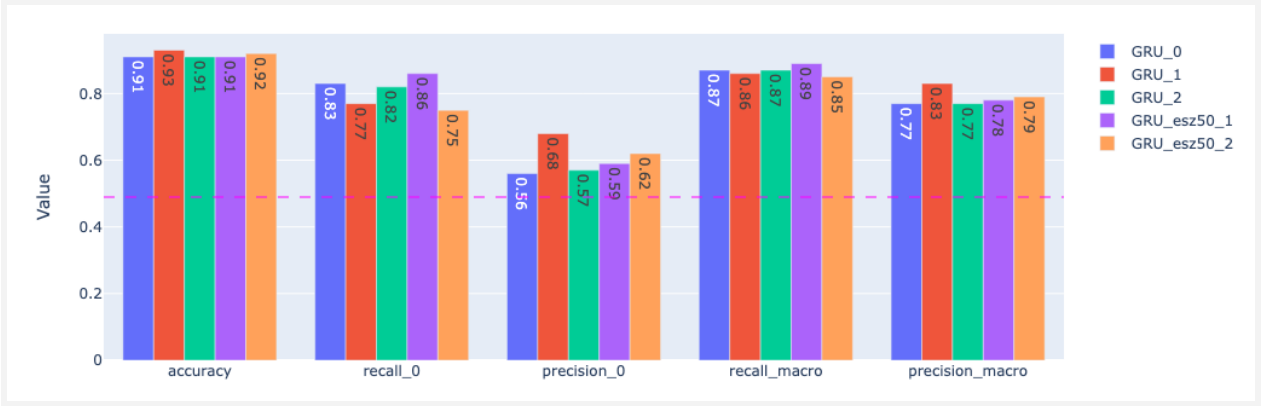
*l'entraînement).* Nous avons joué sur les paramètres class_weight en attribuant plus de poids à la classe du sentiment négatif.

## Résultats des modèles GRU

Graphique 32 - radar des performance des modèles GRU



Graphique 33 - Group barplot - Performance des modèles GRU




**Les modèles GRU testés sont capables de prédire le sentiment correctement dans au moins 90% des cas**. De plus, ils sont bien plus précis dans la prédiction de sentiments négatifs. En effet, lorsqu'ils prédisent un sentiment négatif, ils ont raison dans plus de 56% des cas voir 68% des cas pour le modèle le plus précis(GRU_1). Mais c'est au prix d'une moindre performance dans la capacité à capter les sentiments négatifs.

## Classification multi-classes

Les performance de précision des modèles GRU dans prédictions des sentiments négatifs nous a encouragé à **examiner la problématique de classification multi-classes** où la variable cible est :

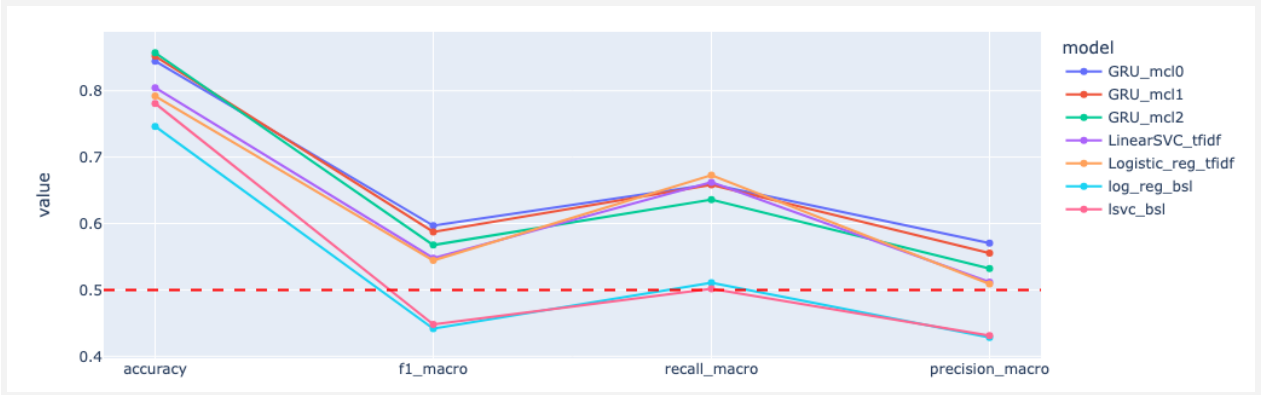
$$sentiment = -1 \ si\, rating < 3$$

$$sentiment = 0 \ si\, rating = 3$$

$$sentiment = 1 \ si\, rating > 3$$

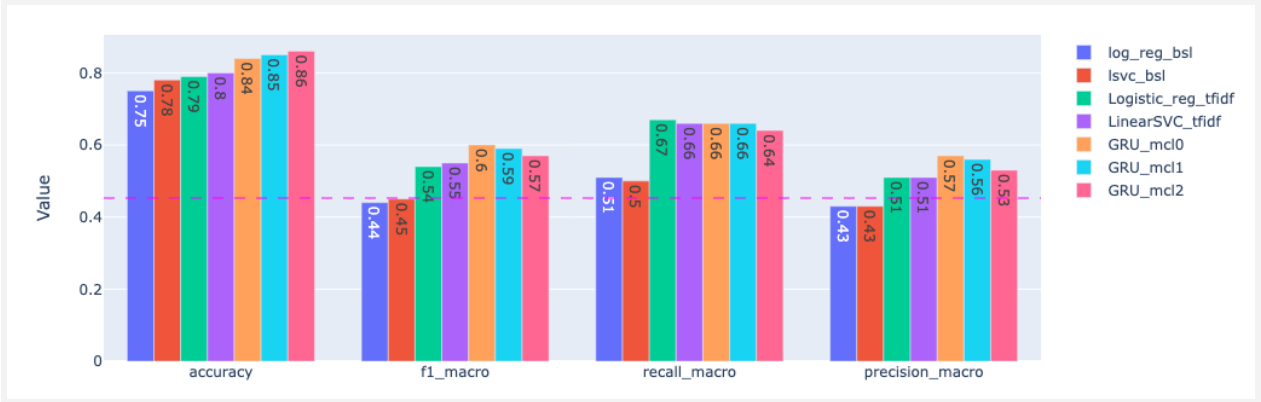
Le graphique ci-après donne un 1er aperçu des performances des différents modèles:

Figure 34 - Métriques-macro des modèles




Log_reg_bsl et lsvc_bsl correspondent respectivement aux modèles de régression logistique et linear SVC de la modélisation baseline initiale.

Figure 35 - Barplot Métriques -macro des modèles




## Analyse de résultats

**La performance des modèles de base( log_reg_bsl et lsvc_bls) est faible avec f1-macro à 45%. Ces modèles ont du mal à classer correctement les sentiments négatifs et neutres.** Cf. la matrice de confusion et le rapport de classification ci-dessous.

Figure 36 -Rapport de classification et Matrice de confusion régression logistique baseline

```
********** log reg **********
              precision    recall  f1-score   support

          -1       0.22      0.43      0.29       368
           0       0.12      0.33      0.17       307
           1       0.95      0.79      0.86      5193

    accuracy                           0.74      5868
   macro avg       0.43      0.52      0.44      5868
weighted avg       0.86      0.74      0.79      5868
```

| Prediction | -1 | 0 | 1 |
|---|---|---|---|
| **Real** | | | |
| **-1** | 158 | 104 | 106 |
| **0** | 95 | 102 | 110 |
| **1** | 460 | 656 | 4077 |

**Les modèles Bag of Words ont une meilleure recall_macro que les modèles GRU**. Cela indique qu'ils sont capables d 'identifier plus de sentiments négatifs ou neutres correctement. **Par contre, comme dans la classification binaire, ils sont moins précis dans la prédiction des sentiments négatifs ou neutres.**

Figure 37 -Rapport de classification et Matrice de confusion régression logistique Tfidf

```
---------------Logistic_reg----------------
```

| Prediction | -1 | 0 | 1 |
|---|---|---|---|
| **Real** | | | |
| **-1** | 221 | 133 | 12 |
| **0** | 106 | 174 | 26 |
| **1** | 339 | 604 | 4056 |

```
              precision    recall  f1-score   support

          -1       0.33      0.60      0.43       366
           0       0.19      0.57      0.29       306
           1       0.99      0.81      0.89      4999

    accuracy                           0.78      5671
   macro avg       0.50      0.66      0.54      5671
weighted avg       0.91      0.78      0.83      5671
```

**Les modèles GRU affichent une meilleure précision d'ensemble**, f1-macro allant de 57% à 60% et la précision macro est aussi meilleure au-delà de 53%. **Ils sont plus précis dans la prédictions du sentiment négatif ou neutre. Néanmoins ils captent moins de sentiments négatifs ou neutres que les modèles Bag of Words.**

Figure 38 - Modèle GRU_mcl0: matrice confusion et rapport de classification

| Prediction<br>Real | 0 | 1 | -1 |
|---|---|---|---|
| **0** | 123 | 52 | 71 |
| **1** | 304 | 3725 | 99 |
| **-1** | 104 | 21 | 169 |

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.23 | 0.50 | 0.32 | 246 |
| 1 | 0.98 | 0.90 | 0.94 | 4128 |
| -1 | 0.50 | 0.57 | 0.53 | 294 |
| accuracy | | | 0.86 | 4668 |
| macro avg | 0.57 | 0.66 | 0.60 | 4668 |
| weighted avg | 0.91 | 0.86 | 0.88 | 4668 |

## Comparaison Modèle de base versus Bags of Words versus GRU

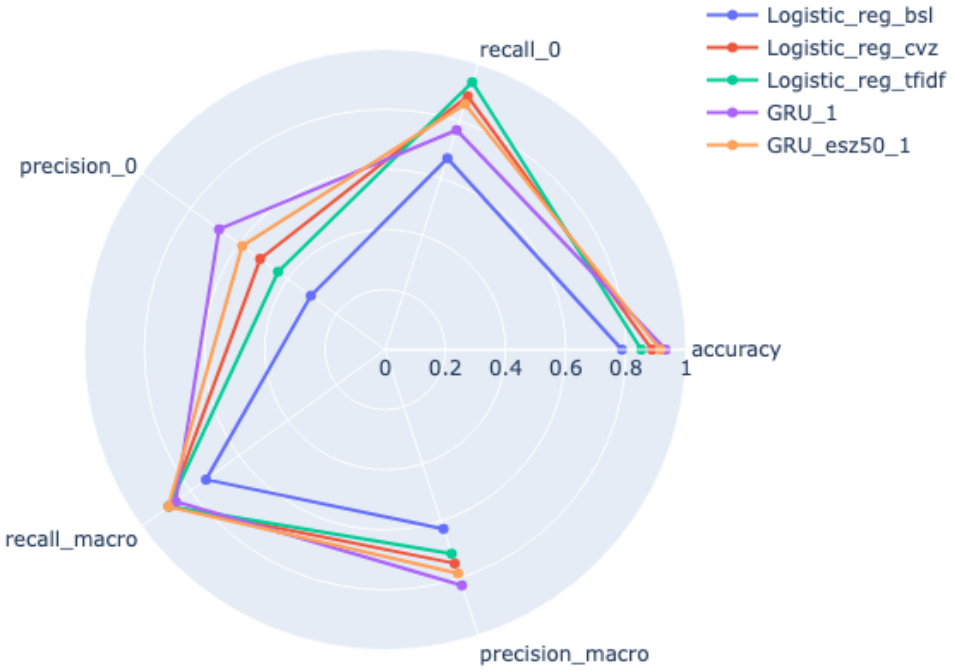
Pour prédire le sentiment positif ou négatif à partir des commentaires des clients, nous nous sommes basés dans un premier temps sur trois variables explicatives uniquement: la longueur du titre du commentaire, la longueur du texte du commentaire et la durée entre la date de transaction et la création du commentaire.

Ensuite nous avons appliqué le **modèle Bag of Words avec Count Vectoriser et TfidfVectorizer** sur des commentaires préalablement nettoyés, lemmatisés et racinisés. Au passage , en plus des modèles traditionnels de classification binaire, nous avons aussi appliqué **un réseau de neurones dense simple à deux couches**.

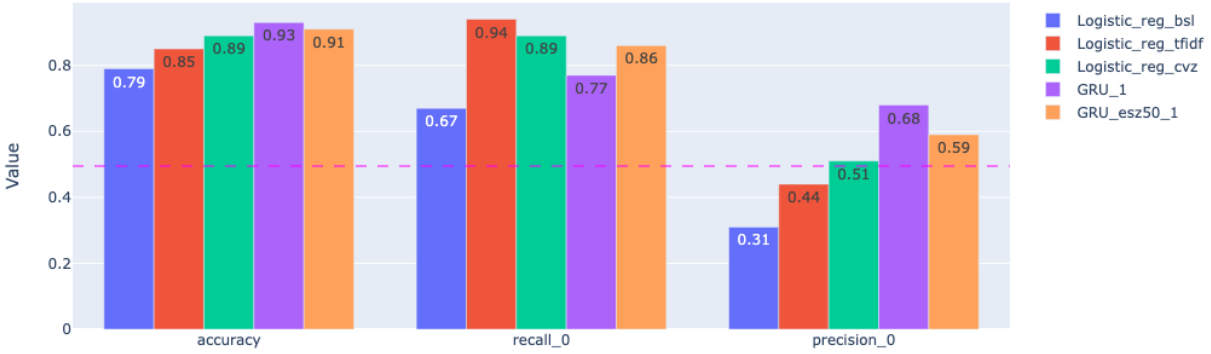
Enfin nous avons appliqué **un réseau de neurones récurrent GRU** pour tenter d'améliorer la qualité des prédictions.

Pour mettre en exergue les performances des 3 approches nous prenons la régression logistique comme modèle de référence.

Graphique 39 - Radar - Comparaison des métriques




Graphique 40 - barplot - Comparaison des métriques




## En conclusion,

La modélisation de base de départ avec uniquement 3 variables explicatives nous permettait déjà de prédire correctement 79% des sentiments et de capter 67% des sentiments négatifs. Par contre près de 70% de sentiments négatifs qu'elle prédit sont en réalité des sentiments positifs.

La modélisation Bag of Words avec Tfidf et CountVectorizer nous a permis d'améliorer significativement les performances des prédictions, avec notamment une précision d'ensemble portée jusqu"à 89%, et surtout une bien meilleure capacité à capter les sentiments négatifs. En effet, la régression logistique de base associée à Tfidf est capable de capter 94% des sentiments négatifs. Nous avons constaté une nette amélioration dans la précision de la prédiction du sentiment négatif par rapport au modèle de base. Même si cette précision atteint 51%, elle reste faible puisque près de la moitié des sentiments négatifs prédits sont en réalité des sentiments positifs.

L'on peut penser que la faible précision dans la prédiction du sentiment négatif est dû au fait que **le "Bag of Words" ne tient pas compte de l'ordre des mots et perd donc une information importante relative au sens des commentaires**.

Pour pallier cette limitation du Bag of Words, nous avons mis en œuvre des réseaux de neurones Gated Récurrent Units(GRU). **Les GRU sont adaptés au traitement de données séquentielles, et sont capables de prendre en compte les informations contextuelles passées lors de la prise de décision pour une tâche.**

Comme le montre les graphiques 39 et 40, **les modèles GRU ont une meilleure précision d'ensemble, au-dessus de 90%. Ces modèles sont de loin bien plus précis lorsqu'ils prédisent les sentiments négatifs**. En effet le GRU_1 par exemple atteint une précision de 68% dans la prédiction du sentiment négatif, i.e. seulement 32% des sentiments négatifs prédits sont de faux sentiments négatifs. Par contre dans les modèles que nous avons testé, ce qu'on gagne en précision dans la prédiction du sentiment négatif, on le perd un petit peu dans la capacité à capter les sentiment négatifs. Les modèles Bag of Words testés sont plus performants pour capter les sentiments négatifs bien que moins précis.

## Analyse de sentiments

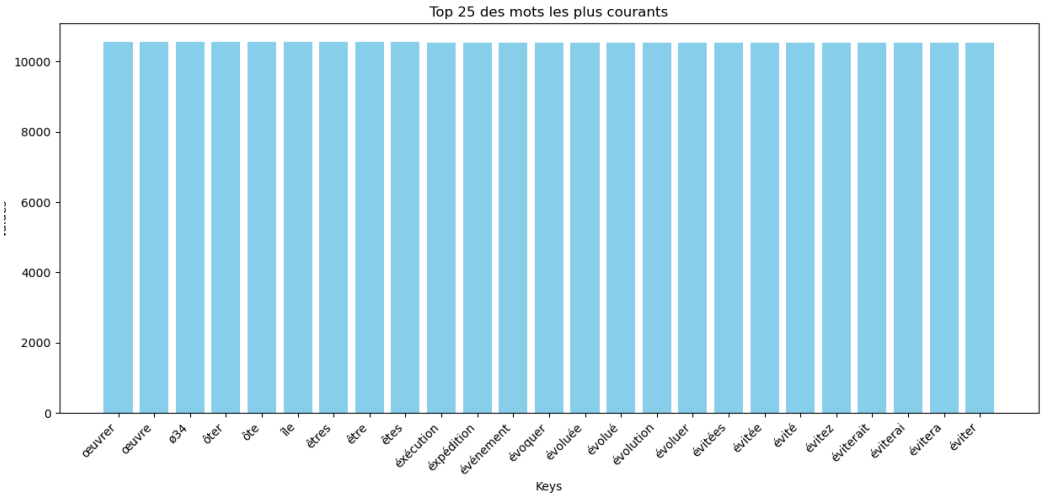
### Preprocessing & Features engineering:

Il est important avant de conduire ce type d'analyse de préparer le texte afin qu'il soit le plus exploitable possible aux différents outils disponibles.

En résumé, nous avons fait en sorte :

- D'**enlever les stop words** qui n'apportent pas de valeur à l'analyse de texte,
- De **tokenizer les commentaires** et de les transformer en liste de mots,
- De **lemmatiser/raciniser les mots** et ainsi réduire la "dimension" de notre texte. En effet, nous avons constaté beaucoup de mots similaires dans les commentaires que nous avons trouvé judicieux de regrouper pour aussi faciliter les étapes de machine learning qui suivront.

L'exemple ci-dessous est très parlant, sans lemmatisation/racinisation, se retrouvent dans le top 25 des mots les plus utilisés i.e avec une occurrence supérieure à 10 000, toute une suite de mots ayant la même racine 'évit'. On peut donc résumer en une racine, 8 mots différents.

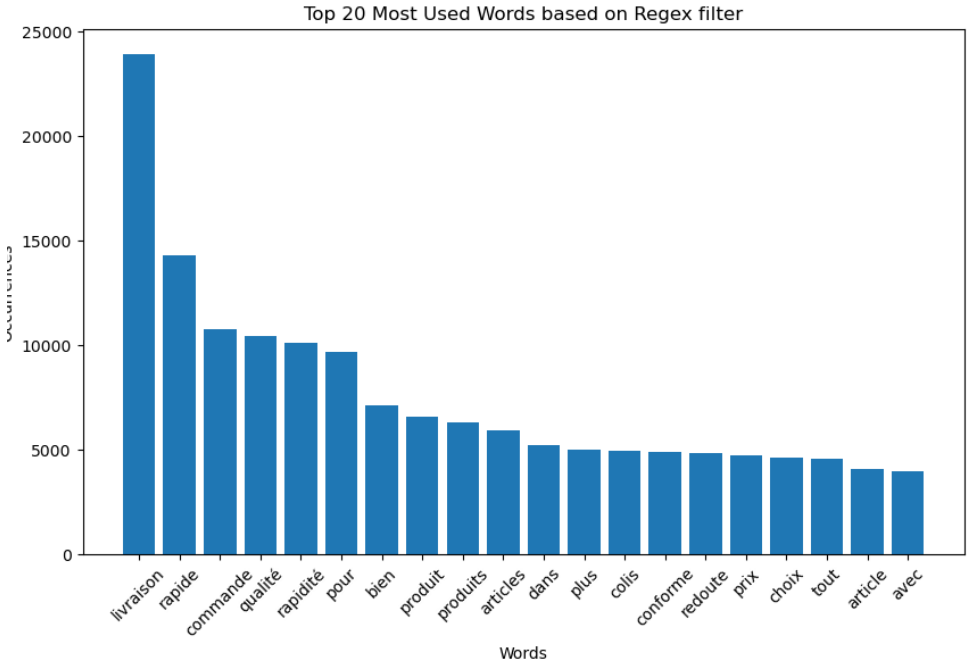



**A posteriori, nous constaterons que la lemmatisation/racinisation n'est pas toujours une étape obligatoire pour obtenir les résultats les plus exploitables possibles.**

- Il est aussi important de **supprimer la ponctuation inutile, de convertir en minuscules et de supprimer les caractères spéciaux** pour plus d'interprétabilité. Dans ce cas, les expressions régulières - entre autres - sont bien utiles.

  Prenons l'exemple ci-dessous, en ayant créé une expression régulière qui filtre sur les mots d'au moins de 4 lettres, **on peut déjà cerner simplement quels sont les sujets récurrents mentionnés par les clients**. En l'occurrence, la livraison, la

rapidité et la qualité ressortent avec une occurrence très importante. Pour rappel, nous travaillons avec un jeu de données d'environ 27000 commentaires.




## Méthodes étudiées :

Il existe différentes méthodes d'analyse de sentiments, plus ou moins complexes, énumérons ci-dessous celles que nous avons développées, de la plus "naïve" basée sur des lexiques de mots à la plus élaborée ou l'on fait appel à des transformers. Il faut aussi noter qu'une analyse de sentiments peut être relativement binaire avec un sentiment positif ou négatif ou plus élaborée avec différentes classes.

**Word Cloud de mots :** dans pratiquement toutes nos analyses et à différents stades de traitement du texte, nous avons généré des nuages de mots pour essayer d'identifier les thèmes récurrents. Que ce soit dans la dissociation des ratings (>=4 et <3) ou dans nos analyses de sentiments ( négatif, neutre, positif).

**Text Blob** : Text Blob est une bibliothèque Python pour le traitement des données textuelles. Elle fournit une API simple pour réaliser les tâches courantes de traitement du langage naturel (NLP) telles que l'étiquetage des parties du discours, l'extraction de phrases nominales, l'analyse des sentiments, la classification, la traduction, etc.

- Objectifs: L'objectif est de prédire de manière non supervisée le sentiment de chaque commentaire à l'aide de Text Blob et de comparer avec les notations des clients.
- Méthodologie: Nettoyage préliminaire du texte des commentaires, élimination des stop words et lemmatisation avec la bibliothèque Spacy qui propose un Lemmatizer

pour le Français. Nous avons ensuite appliqué la fonction d'analyse de sentiments au commentaires nettoyés et aux commentaires lemmatisés.

Lorsqu'il évalue le sentiment d'un texte, Text Blob calcule un score qui est positif si la polarité du sentiment est positive et négatif si la polarité du sentiment est négative.

- Résultats:

La polarité de 73% des commentaires correspond à la notation des clients. C'est-à-dire que si la note est supérieure ou égale à 4, le commentaire est positif, et si elle est inférieure ou égale à 3, le commentaire est négatif. TextBlob détecte que 58% des commentaires négatifs ont une polarité négative. Cependant, parmi les commentaires que Text Blob identifie comme ayant une polarité négative, seulement 23% ont une note négative.

En examinant les commentaires, on trouve des anomalies. Par exemple, certains commentaires avec une note de 4 ou 5 ont un ton négatif. À l'inverse, certains commentaires avec une note négative ont un ton positif. De plus, certains commentaires présentent à la fois une polarité positive et négative. **Ces nuances,ambiguités sont difficilement prises en compte par textBlob**. Ci-après la matrice de confusion et le rapport classification permettant de comparer les prédictions de TextBlob à la cible notation positive ou négative.

*Matrice de confusion :TextBlob versus Notation Client*

| Textblob | 0 | 1 |
|---|---|---|
| Rating | | |
| 0 | 1553 | 1147 |
| 1 | 5223 | 15546 |

```
**** Classification report ****
              precision    recall  f1-score   support

           0       0.23      0.58      0.33      2700
           1       0.93      0.75      0.83     20769

    accuracy                           0.73     23469
   macro avg       0.58      0.66      0.58     23469
weighted avg       0.85      0.73      0.77     23469
```

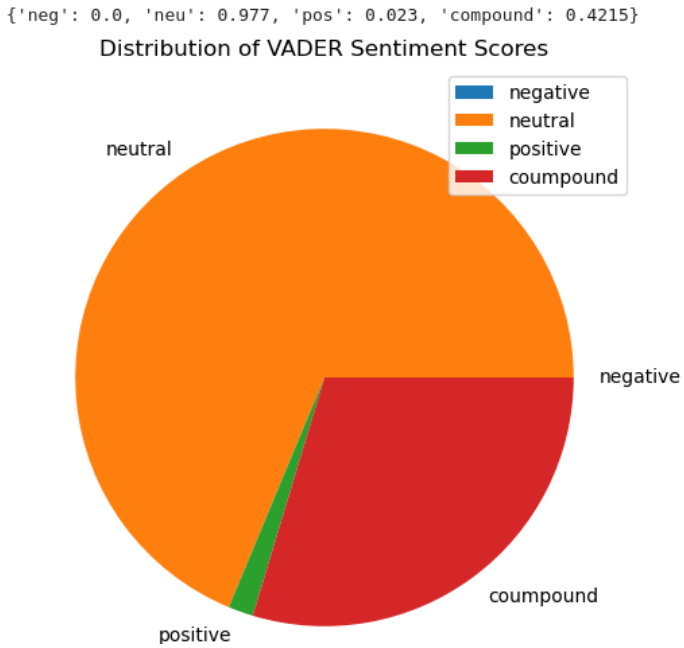
**VADER** : Valence Aware Dictionary and sEntiment Reasoner, VADER utilise un lexique pré-étiqueté qui prend en compte la polarité, l'intensité et les émotions liées aux mots spécialement conçu pour gérer les nuances du langage naturel, y compris les émotions, l'ironie et les sarcasmes, pour une évaluation globale du sentiment,

- Objectifs: l'idée ici était de se familiariser avec le dictionnaire Vader et de faire une analyse de distribution de sentiments à travers les commentaires des clients, et de coupler cette analyse aux bi et trigrams les plus représentés à la fois en sentiment positif qu'en sentiment négatif.

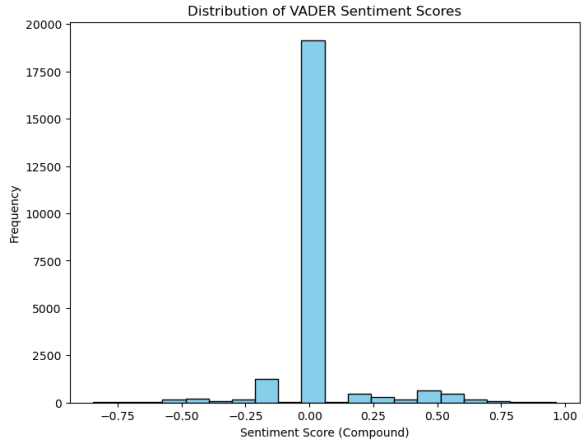
- Méthodologie: utilisation du Sentiment Intensity Analyzer avec analyse dans un premier temps d'un commentaire spécifique pour comprendre comment l'outil réagit, ensuite nous avons effectué l'analyse sur l'ensemble du dataframe et enfin fait du preprocessing de texte avant de générer les n grams. Le but étant d'avoir une représentation graphique de la distribution de sentiments et du sentiment en fonction du ngram. Les n grams choisis sont de 2 et 3 mots regroupés ensemble..
- Résultats :

  Au cas par cas, exemple pris sur un commentaire client : *""je commande depuis longtemps chez la redoute; j'y apprécie le choix , la rapidité de livraison et la facilité de règlement grâce à la carte ,même si presque à chaque fois je demandais un règlement comptant cela m'évitait de laisser mon numéro de carte bleue sur le web ...par contre depuis quelques jours cela a changé je n'y comprends plus rien je ne peux plus payer comme ça alors que je n'ai absolument aucun débit en attente sur mon compte ....du coup par sécurité face à l'internet j'ai dû régler mes deux dernières commandes avec une carte bancaire à usage unique ce qui est un peu plus compliqué pour moi et risque forcement de me faire hésiter pour des futurs achats sur ce site....dommage."*

Distribution du  score de sentiment :




Si on s'attache à un commentaire particulier, **nous comprenons facilement le résultat de l'analyse**. En effet, le client reste satisfait et a donné une note de 4. Sinon, un grand sentiment de neutralité ressort sur cette analyse Vader, surtout parce que les commentaires sont souvent mitigés, en commençant par énumérer une expérience positive avant de changer de ton pour exprimer une expérience négative.




- **Analyse des bigrams et trigrams :**

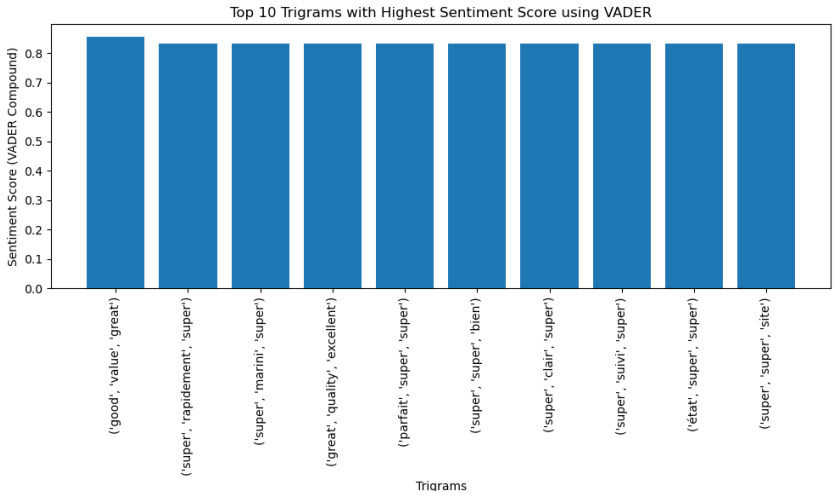
Objectifs : l''analyse des n-grams est une technique couramment utilisée dans le traitement du langage naturel (NLP) et la recherche d'informations. Elle consiste à

diviser un texte en séquences de n mots consécutifs. Les n-grams sont utilisés pour capturer des informations sur la structure et la sémantique du texte. Ils peuvent être utilisés dans des tâches de traitement de texte telles que la catégorisation de texte, la détection de spam, l'analyse de sentiments.

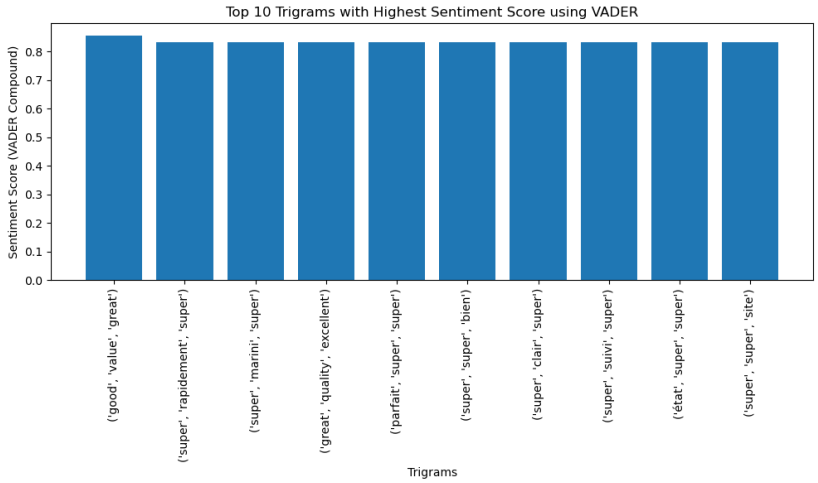
Méthodologie **: dans notre cas, nous avons combiné notre analyse de sentiments VADER avec en entrée des bi et tri grams.** Pour rappel, les n grams ont été générés après un preprocessing du commentaire ( tokenisation, stop words et utilisation de regex)

Résultats :

Top 10 bigrams avec le plus haut score de sentiment : on constate ici que pour les commentaires positifs, les bi grams ne ressortent pas beaucoup de valeur ajoutée hormis une appréciation positive des clients vis à vis de la Redoute.

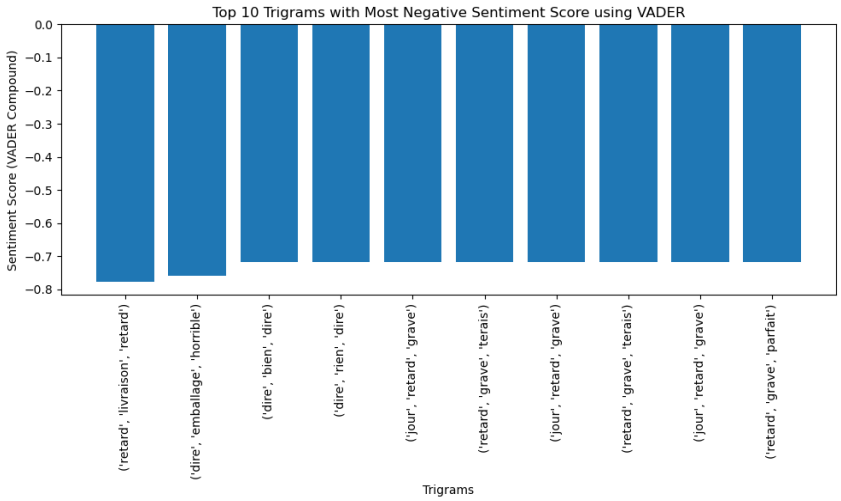



Top 10 trigrams avec le plus haut score de sentiment : les trigrams sur un score de sentiment positif apportent à peine plus, on commence à percevoir certaines notions liées à la rapidité, la qualité des produits mais pas encore assez de matière pour exploiter l'information en tant que telle.




Top 10 trigrams avec le plus bas score de sentiment: sûrement l**a plus intéressante des 3 analyses** quant à identifier les points d'amélioration.

En effet ici **on y retrouve des mots à plus grande valeur ajoutée, tels que "retard livraison", "emballage horrible"** mais pas encore assez de matière pour faire une vraie sérialisation des problématiques remontées par les clients.




**Multinomial Naive Bayes** : il s'agit d'un algorithme de classification bayésienne naïve qui est adapté pour des données catégorielles discrètes.Il est spécifiquement conçu pour traiter des données catégorielles, généralement représentées par des comptages ou des fréquences, comme c'est souvent le cas dans le cas du traitement du langage naturel où l'on compte les occurrences de mots. Il **repose sur l'hypothèse naïve d'indépendance conditionnelle entre les caractéristiques,** ce qui signifie que l'occurrence d'une caractéristique particulière n'est pas influencée par l'occurrence des autres caractéristiques, ce qui simplifie le calcul des probabilités.

- Objectifs: tester ce modèle en le pré-entraînant manuellement càd que nous avons échantillonné 50 commentaires et labellisé chaque commentaire pour en ressortir un sujet dominant.

  Voici un exemple des labels créés:

  - ☐ Retard_livraison,
  - ☐ Retour_colis,
  - ☐ Service_client,
  - ☐ Qualité,
  - ☐ Paiement,
  - ☐ Livraison_partielle,
  - ☐ Article_non_conforme,
  - ☐ Prix,
  - ☐ etc.

- Méthodologie: Tout d'abord, nous avons nettoyé nos commentaires avec des stop words et une expression régulière pour extraire les mots qui sont représentatifs et d'au moins 4 lettres. Ensuite, nous avons créé une nouvelle colonne "label" avec les commentaires labellisés à la main et **entraîné le modèle Naive Bayes avec comme features X, les commentaires clients et comme cible le label** puis prédit la colonne label sur l'ensemble du jeu de données. Avant d'entraîner le modèle, nous sommes passés par une **étape de classification de texte via TF-IDF pour vectoriser et identifier les mots les plus importants**

  **Nous avons dans cette analyse ressorti les bi-grams et tri-grams** en tentant de voir s'il y avait une relation avec le sujet du commentaire, ce qui n'est pas le cas car le n_gram n'est qu'un filtre qui passe sur le texte pour regrouper les mots.

- Résultats: le résultat n'est pas concluant, **il aurait sûrement fallu labelliser plus de commentaires, 50 lignes sur 27000 lignes reste un échantillon très faible**. On constate quand même que l**e modèle est plus sensible aux problèmes de qualité et de livraison qui s'avèrent des sujets récurrents dans les commentaires clients.**

**BERT** (Bidirectional Encoder Representations from Transformers) est un modèle de traitement du langage naturel (NLP) basé sur des transformers, qui sont des modèles de réseaux de neurones particulièrement performants dans le traitement séquentiel de données. L'utilisation de BERT pré-entraîné pour l'analyse de sentiment est puissante car il peut comprendre le contexte et les relations entre les mots de manière plus sophistiquée que les méthodes traditionnelles. Cependant, l'utilisation de BERT peut nécessiter des ressources informatiques importantes en raison de sa complexité.

- Objectifs : vérifier la pertinence d'un modèle pré-entraîné et basé sur une technologie Transformers par rapport a Vader qui reste un modèle plus simple et moins consommateur de ressources.

  En effet, Vader est basé sur des règles et utilise un lexique pré-annoté de mots avec des scores de polarité pour déterminer le sentiment d'un texte. Il ne capture pas les relations contextuelles de la même manière que les modèles basés sur des transformers.

- Méthodologie : nous avons ici aussi utilisé les outils habituels de nettoyage de texte comme les stop_words, la suppression de la ponctuation et des chiffres, et appliqué une expression régulière pour ne ressortir que les mots de plus de 4 lettres.
- Résultats : L'analyse BERT nous révèle une évaluation de sentiments sur 4 niveaux : 0,1,2 et 3 qui ne correspondent pas directement aux ratings clients allant de 1 à 5. Cependant nous avons constaté **une meilleure corrélation analyse de sentiments/rating client.**

## Prédiction de la réponse fournisseur en fonction de la note du client:

**Tests statistiques**

Notre premier objectif est de valider statistiquement la relation entre le rating et la réponse fournisseur.

Nous avons passé plusieurs tests pour identifier la corrélation entre ces variables.

L'approche adoptée a consisté dans un premier temps à une vérification des suppositions:

- 1/ Homogénéité des variances : nous avons appliqué ici un test de LEVENE
- 2/ Normalité des résidus : Une visualisation fait via un QQ-plot

Le 2eme volet consiste à entamer les tests statistiques les plus adaptés.

***1/ Homogénéité des variances / Test LEVENE***

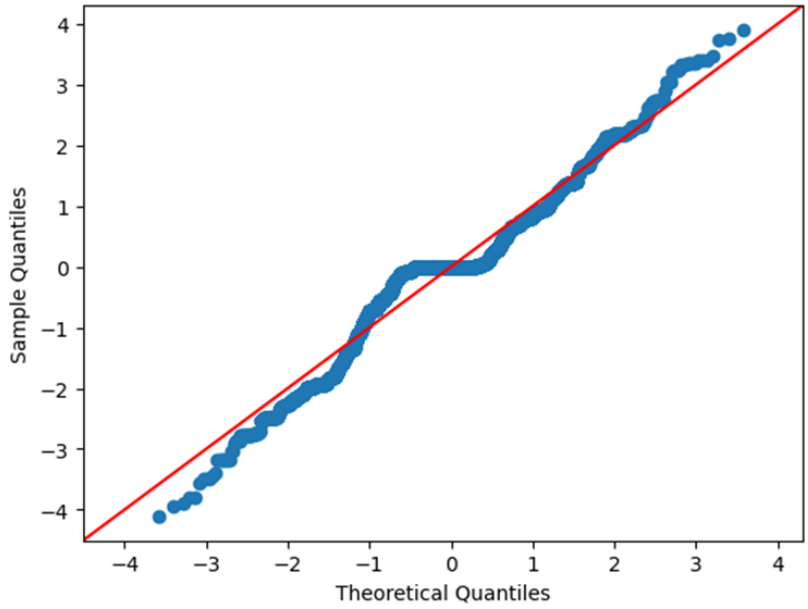
Output

```
Statistique: 1.72604153619374
P-valeur: 1.093552110575327e-38
```

Une p-valeur aussi petite indique que l'on peut rejeter l'hypothèse nulle d'égalité des variances entre les différents groupes (chaque groupe correspond à une réponse de « SupplierReply »). Donc les données suggèrent qu'il existe des différences significatives dans les variances des notes pour chaque type de réponse prédéfinie du fournisseur.

***2/ Normalité des résidus***

Graphique




Dans notre graphique, la plupart des points semblent suivre la ligne rouge, mais avec quelques écarts aux extrémités, indiquant des queues plus lourdes que celles de la distribution normale. Cela signifie que les résidus sont en grande partie normalement distribués, mais il existe des preuves de kurtosis (c'est-à-dire des pics plus prononcés ou des queues plus lourdes que celles d'une distribution normale)

**Nous avons opté pour un test moins sensible à la normalité et à l'égalité des variances. Notre choix s'est porté sur le test non paramétrique de KRUSKAL-WALLIS.**

## Test de KRUSKAL-WALLIS :

Output :

```
Statistique de test: 3731.4976588010613
P-value: 3.268815160296018e-225
On rejette l'hypothèse nulle : il existe des différences significatives entre les groupes.
```

Avec une valeur de p aussi petite, on a une très forte évidence statistique que la distribution des notes n'est pas la même pour tous les types de réponses prédéfinies (SupplierReply).

Dans la littérature on lit que le test de KRUSKAL-WALLIS ne nous indique pas quels groupes sont différents ou la nature de ces différences il est conseillé d'envisager de faire des analyses post hoc avec des tests par paires pour comparer les distributions des notes entre

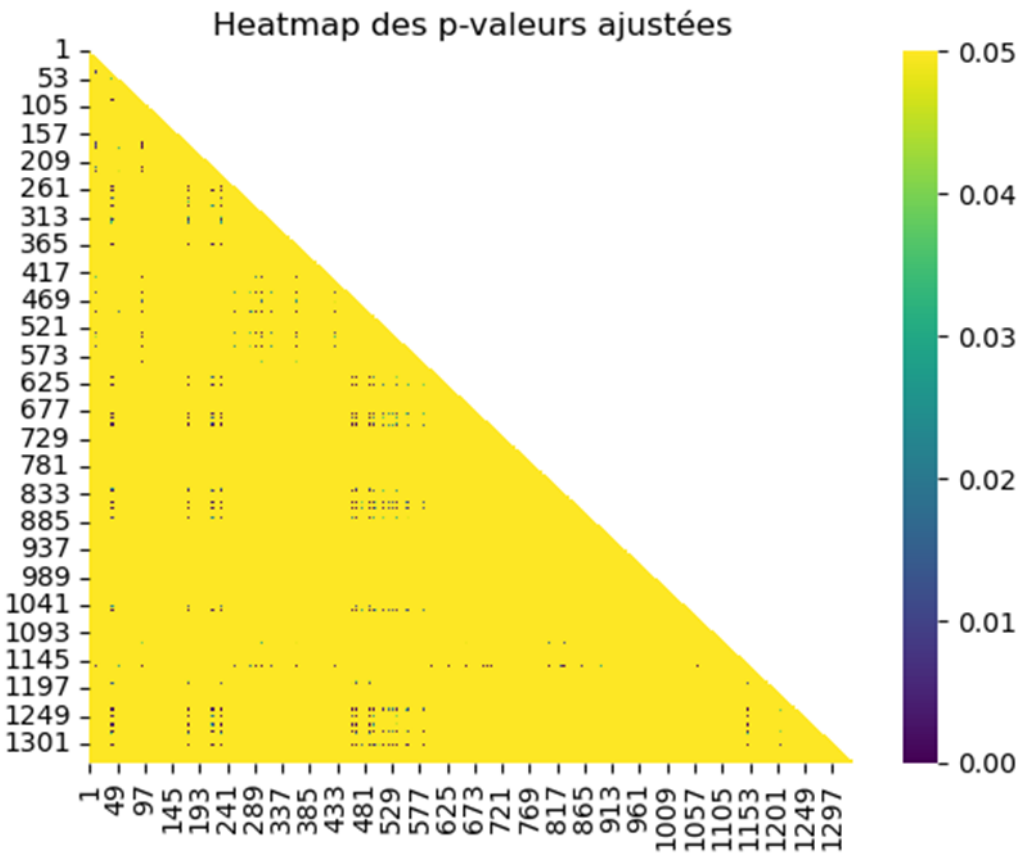
chaque paire de groupes. Une méthode courante est le test de Dunn pour les comparaisons multiples après un test de KRUSKALL-WALLIS ce que nous allons entamer.

## Test de DUNN

Dans ce test, nous avons dû modifier la méthode d'ajustement à 3 reprises pour confirmer le test. Nous avons d'abord ajusté avec la méthode Bonferroni mais la matrice en sortie était emplie de « 1 ». Nous avons ensuite essayé d'ajuster sans méthode pour vérifier la matrice en sortie. Nous avons obtenu un résultat. Enfin, nous avons utilisé la méthode d'ajustement « fdr_bh ».

Nous avons effectué la visualisation des résultats via un heatmap / paires avec p-values < 0,05.

Graphique



Synthèse des tests statistiques

Significativité Statistique : L'application du test de Dunn avec ajustement 'fdr_bh' a révélé des différences significatives entre plusieurs groupes, **indiquant une corrélation potentielle entre les variables étudiées.**

Pertinence des Données : La vérification préalable de l'homogénéité des variances et de la normalité des résidus **confirme la fiabilité des données,** ce qui rend la modélisation plus pertinente pour extraire des insights significatifs.

Potentiel de Découverte : Les résultats obtenus fournissent une base solide pour une exploration plus approfondie à travers la modélisation, permettant de mieux comprendre les dynamiques sous-jacentes et d'optimiser les prises de décisions basées sur les données.

## Premières modélisations avec analyse de sentiment

Démarche

Nous avons suivi une approche graduelle pour modéliser les données. Nous **avons privilégié principalement l'utilisation de modèles non supervisés.**

1/ MODELISATION TEXT MINING

La première modélisation est une classification de commentaires clients utilisant le traitement du langage naturel et l'apprentissage automatique.

Elle inclut le nettoyage des données, l'élimination des mots superflus, la **conversion en vecteurs via TfidfVectorizer, et l'application d'une régression logistique.**

Les données sont divisées en ensembles d'entraînement et de test pour évaluer le modèle, qui est mesuré par son exactitude et une matrice de confusion.

Output MODELISATION TEXT MINING

```
Accuracy: 0.5456110154905336
[[ 19  14 160   6   0]
 [ 18   5 149  10   0]
 [ 20   8 255  35   0]
 [  3   1  73 355   0]
 [  0   2   6  23   0]]
```

**La précision du modèle est de 54.6%, ce qui indique une performance modeste**. La matrice de confusion montre des erreurs significatives de classification entre les classes, suggérant des difficultés du modèle à différencier correctement entre les différentes notes de commentaires.

2/ MODELISATION VADER

Nous allons **utiliser le « Sentiment Intensity Analyzer » de NLTK** pour obtenir les « neg/neu/pos scores » du texte. On utilise la technique du « bag of words » :

1.Les Stop words sont retirés

2. Chaque mot est "scoré" et ensuite combiné avec le score total.

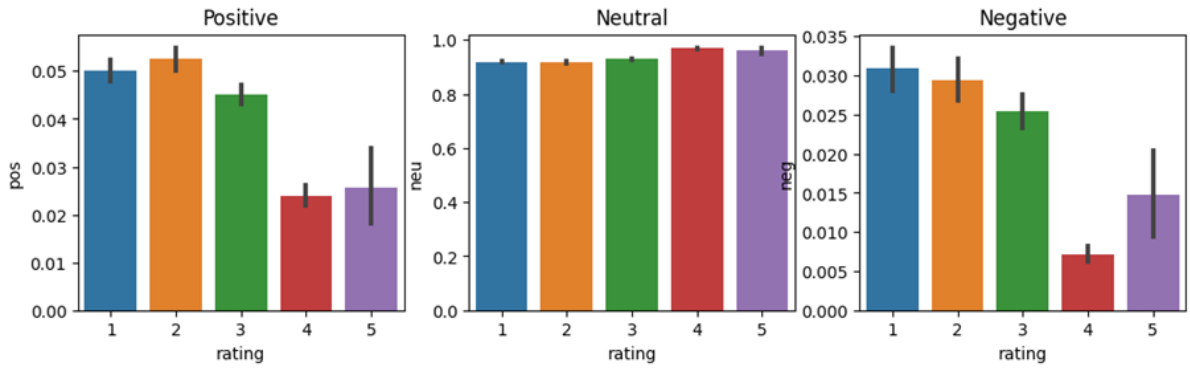
Pour faciliter leur analyse, nous les **présenterons sous forme d'un nouveau DataFrame avec 3 colonnes qui représentent le type de sentiment** : neg/ neu / pos est mis en miroir avec le rating client.

Graphique

| | id | neg | neu | pos | compound | rating | comment |
|---|---|---|---|---|---|---|---|
| 0 | rev-0134d691-cf06-4caa-8d42-528cea655fa2 | 0.0 | 0.9 | 0.1 | 0.5696 | 4 | Une livraison assez rapide, les tailles sont b... |
| 1 | rev-97bff9c6-1d08-4caf-8c8a-5fc8f168d817 | 0.0 | 1.0 | 0.0 | 0.0000 | 4 | Juste vous dire que la qualité est inférieure ... |
| 2 | rev-b9f9a2c6-9c55-48d4-980a-3224367b0fc8 | 0.0 | 1.0 | 0.0 | 0.0000 | 4 | Vendeur et vendeuse au top dans le magasin Gal... |
| 3 | rev-31ced4d5-10da-49dc-9111-8eb9ee5f2fe4 | 0.0 | 1.0 | 0.0 | 0.0000 | 4 | M'ayant été signalé que mon colis ne pouvait ê... |
| 4 | rev-d1d74f83-bd9a-468f-beeb-b2c64ebc65b2 | 0.0 | 1.0 | 0.0 | 0.0000 | 4 | J'apprécie la redoute pour sa fiabilité et son... |

Nous avons réalisé quelques représentations graphiques pour visualiser de manière claire et intuitive les résultats obtenus avec le modèle VADER.

Visualisation de la relation sentiment pour « pos », « neu », « neg » et rating :




Les graphiques montrent les scores agrégés par note d'évaluation, avec **une tendance claire que les scores neutres restent élevés sur toutes les évaluations**. Les scores positifs et les scores négatifs ne suivent pas une corrélation logique.

Cette visualisation du scoring se confirme par **une accuracy décevante à 21%.**

Output:

```
L'accuracy du modèle est: 0.21
```

3/ MODELISATION BERT :

Nous poursuivons nos tests de **modélisation non supervisés avec un modèle BERT** mais pré- entraîné avec le modèle BERT dans le cas présent:

`'nlptown/bert-base-multilingual-uncased-sentiment'`

Après l'installation des différentes librairies et dépendances, nous instancions le modèle et testons une série de phrases. Les resultats sont coherents et **BERT montre une capacité à contextualiser le sens sur des mots mais aussi des groupes de mots.**

```
tokens = tokenizer.encode("je trouve ce produit super et rien a redire sur l'emballage et la livraison sont parfaits.",return_tensors='pt')
```

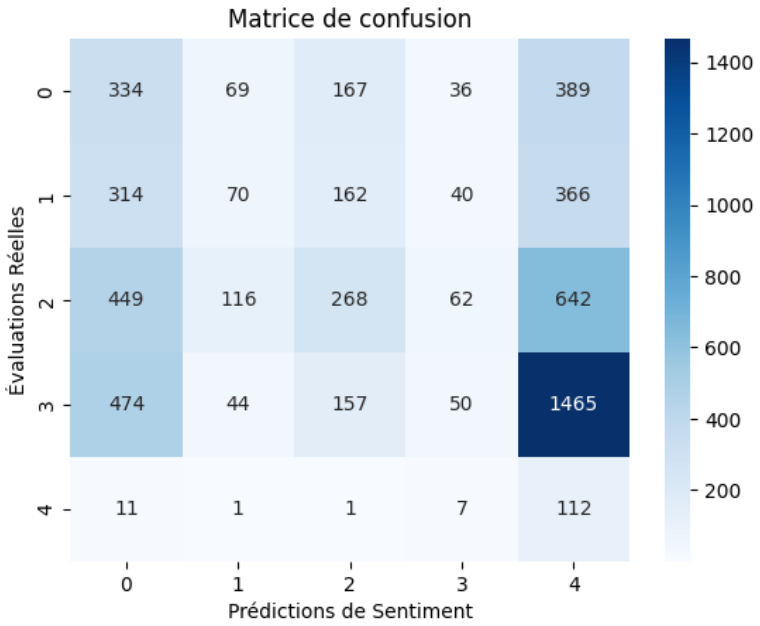
```
result = model(tokens)
```

```
int(torch.argmax(result.logits))+1
```

```
5
```

Nous appliquons le modèle à l'ensemble du DataFrame et nous présentons les résultats via une matrice de confusion.

Matrice de confusion Bert:




Nous calculons également l'accuracy de ce modèle qui tombe à 14%, sachant que nous avons essayé de travailler en approche multi-classes.

Output:

```
L'accuracy du modèle est: 0.14
```

**Les résultats sont moins bons que la modélisation naïve, instanciée au démarrage.Ils ne sont pas performants et nous renforcent dans l'idée d'une neutralité de ton dans les réponses fournisseurs.**

**Les modèles instanciés ne nous permettent pas de prédire en l'état un score de sentiment en correspondance avec le rating.**

Au vu des résultats décevants des premiers jeux de modèles, il devenait indispensable de se replonger dans l'étude des types de phrase disponibles dans la colonne «Supplier Reply ».

## Un retour dans les “Verbatims”

### Démarche

Il en ressort synthétiquement, q**u’après une observation détaillée des verbatims**, **nous avons l'impression qu’en fonction du rating le processing de traitement par le service client est différent.** Par exemple**,** lorsque le rating est égal à 4 ou plus, la réponse semble générique avec quelques variations.

De plus, il semble que même si le rating est supérieur à 3, une analyse de sentiment semble avoir lieu (ou bien un traitement manuel commentaire par commentaire) pour déterminer le type de réponse type à apporter.
D’autant plus que les réponses fournisseurs dans le cas d’un rating entre 1 et 3 sont quasi toujours soit un constat de satisfaction de résolution du problème, soit un constat d’incapacité à joindre le client.
Cela semble suggérer une réponse après un traitement spécifique et personnalisé - au cas par cas- pour les commentaires avec un rating inférieur ou égal à 3.
Nous pensons que la seule singularité se trouve dans le nom de la personne chargée du traitement de la doléance.

Cela pourrait expliquer le ton neutre des réponses et les mauvais scorings de sentiment des modèles testés auparavant.

## Analyse des réponses pour les ratings de 4 ou plus :

Extraction des Données : preprocessing du Texte pour les lignes où le rating est supérieur ou égal à 4.

- Création d'un WordCloud et itérations successives de stop Word amélioré

Graphique 1er WordCloud

Graphique dernier WordCloud

- Génération d' un nouveau WordCloud après avoir supprimé les stop words pour voir si les mots restants confirment notre hypothèse sur le caractère générique des réponses.

Constat

L'analyse de ce WordCloud nous donne une indication initiale sur la nature du texte analysé. Ce dernier semble être générique et nous avons réalisé quelques analyses supplémentaires.

Tests complémentaires :

Analyse de Fréquence des Mots

| | Mot | Fréquence |
|---|---|---|
| 28 | de | 5018 |
| 15 | nous | 3830 |
| 22 | votre | 2586 |
| 29 | service | 2354 |
| 13 | léquipe | 2323 |
| 30 | client | 2315 |
| 0 | bonjour | 2312 |
| 17 | à | 1913 |
| 1 | merci | 1611 |
| 19 | vous | 1522 |
| 33 | le | 1398 |
| 25 | très | 1127 |
| 24 | a | 1108 |
| 26 | vite | 1024 |
| 2 | pour | 944 |
| 60 | sommes | 905 |
| 14 | et | 841 |
| 59 | expérience | 835 |
| 27 | ilham | 784 |
| 35 | partager | 727 |

Analyse bigrammes trigrammes
Output (apercu):

```
[(('service', 'client'), 2314), (('client', 'bonjour'), 2294), (('de', 'léquipe'), 2167), (('léquipe', 'service'), 2166),
[(('service', 'client', 'bonjour'), 2293), (('de', 'léquipe', 'service'), 2165),
```

Diversité Lexicale
Output:

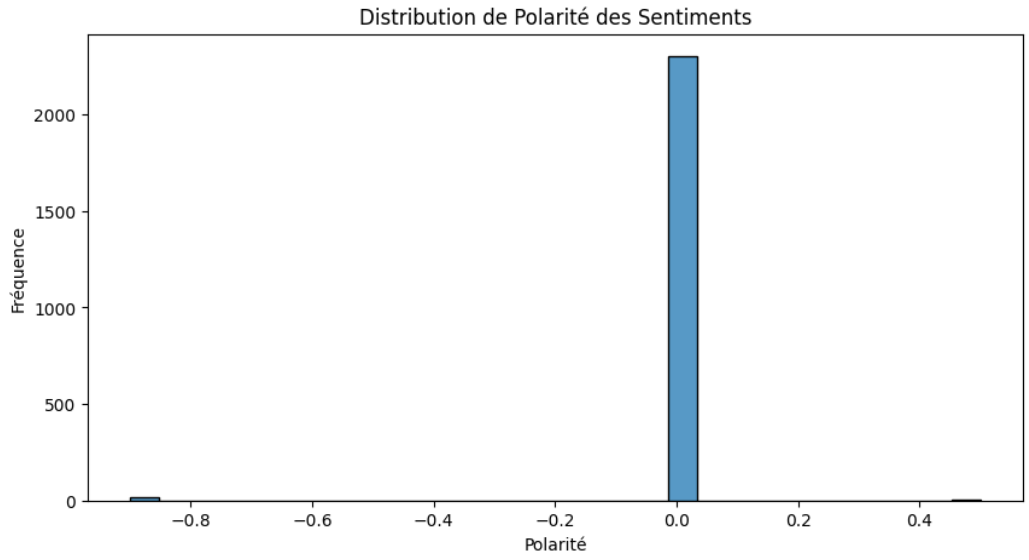
```
Diversité Lexicale: 0.006016525809796098
```

Classification et Clustering de Texte : transformation des données textuelles en TF-IDF et définition du modèle K-Means avec **Affichage des centres de cluster**

```
Cluster 0:
['redoute', 'trouver', 'ensemble', 'ses', 'afin', 'léquipe', 'nouveau', 'lintérêt', 'portez', 'renouvelle']
Cluster 1:
['léquipe', 'quel', 'œuvre', 'fluidité', 'faisons', 'ferons', 'fidélitévos', 'financier', 'grand', 'expertise']
Cluster 2:
['positifs', 'vôtre', 'ravis', 'léquipe', 'oumaima', 'yazid', 'lamia', 'ikram', 'restons', 'aident']
Cluster 3:
['aident', 'améliorer', 'léquipe', 'prises', 'été', 'restons', 'causé', 'transporteur', 'rencontré', 'retard']
Cluster 4:
['restons', 'léquipe', 'quel', 'redoute', 'ikram', 'œuvre', 'financier', 'faisons', 'ferons', 'fidélitévos']
```

Analyse de Sentiment simple "TextBlob"

Graphique




Constat

**La batterie d' analyses complémentaires nous conforte sur la nature du texte analysé. Ce dernier est vraisemblablement générique.**

## Analyse des Réponses pour les Ratings Entre 1 et 3:

Extraction des Données : pré-processing du Texte pour les lignes où le rating est entre 1 et 3.

Analyse de Contenu : Etude du contenu pour identifier les éléments communs et les différences par rapport aux réponses pour les ratings plus élevés.

Nous commençons par l'initialisation et l'application du Vectorizer aux réponses.Puis nous convertissons en Data Frame pour une meilleure lisibilité, pour finir nous affichons les termes les plus fréquents.

Output:

```
trouver        1334
redoute        1308
ensemble       1270
via            1253
afin           1227
avons          1125
améliorer      1059
se             1017
https          1011
laredoute      1011
fr             1011
espace         1011
www            1011
aspx           1010
trouve         1009
navigateur     1009
conseiller     1005
je              986
ses             829
succès          827
```

Identification du Personnel Répondant : nous utilisons des techniques de traitement de langage naturel (NLP) pour extraire les noms des employés mentionnés dans les réponses. Cela nous aide à déterminer si les réponses sont personnalisées.

Output:

| id | SupplierReply | Noms |
|---|---|---|
| rev-102b7ec2-33e3-442d-8e79-72ccea10ff6a | Bonjour, nous avons tenté de vous joindre sans... | [] |
| rev-e31e7a6c-77e7-4ad0-b7d2-b0072deea4f1 | Bonjour, nous sommes heureux d'avoir pu trouve... | [Soukaina] |
| rev-eac63773-8559-43d0-a4ab-88fa5cc32268 | Bonjour, nous sommes heureux d'avoir pu trouve... | [] |
| rev-2e9eb857-a5a8-4fd4-a436-089db3782858 | Bonjour, nous avons tenté de vous joindre sans... | [Yousra de l'] |
| rev-4fe712bd-7451-4232-9d76-f05aaaa538c2 | Bonjour, ravi d'avoir pu vous joindre pour éch... | [] |
| rev-c1f26089-62da-4240-8e2b-b2dd3dc37f17 | Bonjour, nous avons tenté de vous joindre sans... | [Yousra de l'] |
| rev-1df9979a-1ee4-4f11-a1bc-477b698a9208 | Bonjour, nous avons tenté de vous joindre sans... | [Adnane de l'] |
| rev-9af5a3c4-7194-441e-815d-9d211b1271f8 | Bonjour, nous avons tenté de vous joindre sans... | [Adnane de l'] |
| rev-9a287b65-897b-4ac8-8293-34253f09fe12 | Bonjour, nous avons essayé de vous contacter. ... | [Oumaima de l'] |
| rev-8d6e0af5-ee33-418d-b8e3-277d46f52206 | Bonjour, nous avons tenté de vous joindre sans... | [Yousra de l'] |
| rev-72b1f9b8-f09d-4c59-870a-6b306eaa98b3 | Bonjour, merci d'avoir pris le temps de nous p... | [Ilham] |
| rev-5f72a266-595f-488a-a8ff-228bf4ea6403 | Bonjour, nous avons tenté de vous joindre sans... | [Yousra de l'] |
| rev-e309a034-6f44-4084-837d-053563ac5882 | Bonjour, nous sommes heureux d'avoir pu trouve... | [Jamila] |
| rev-e7d0d762-e5d4-4990-a9ee-15eb4d97e77a | Bonjour, merci d'avoir pris le temps de nous p... | [] |
| rev-c38ab27a-e8a6-4631-91a1-5b0b92740f9f | Bonjour, nous avons tenté de vous joindre sans... | [Yousra de l'] |

Pour analyser plus en détail, il nous semble utile de **séparer les réponses attribuées à "Ilham"** qui est un nom qui ressort plus fréquemment dans les commentaires que les autres. Et vérifier si Ilham répond différemment. De plus, nous calculons le nombre total de réponses par intervenant pour révéler si certains membres de l'équipe sont plus engagés que d'autres dans les réponses personnalisées.

Output:

```
Cumul des réponses par intervenant :
Ilham                 603
Vos                   574
Wassim                419
Loubna de l'          314
Zineb de l'           235
Loubna                184
Ilham de l'           148
Soukaina de l'        145
Soukaina              104
Bonne                  89
Lamia                  73
Wassim de l'           72
n'                     59
Adnane de l'           31
Oumaima de l'          31
Yousra de l'           26
```

Cette analyse révèle plusieurs points intéressants :

La présence de noms personnels : "Ilham", "Soukaina", "Yousra", "Loubna", et "Zineb" sont mentionnés très régulièrement, indiquant que certaines réponses incluent des noms d'employés, ce qui pourrait être un **indicateur de personnalisation des réponses.**

L'analyse des noms suggère une tentative de personnalisation, mais la présence récurrente des mêmes noms pourrait également indiquer que ces mentions font partie d'un template de réponse.

Analyse de Sentiment:
Nous avons implémenté une analyse de sentiment Bert pour évaluer le ton des réponses, afin de voir si elles sont plus empathiques ou personnalisées sur des avis positifs.

Nous commençons par charger le pipeline 'sentiment-analysis' avec le modèle multilingue : 'nlptown/bert-base-multilingual-uncased-sentiment'. Ensuite nous définissons une fonction pour obtenir le score de sentiment et nous appliquons la fonction aux réponses.

Nous visualisons ensuite le « sentiment »:

| id | SupplierReply | rating | Sentiment |
|---|---|---|---|
| rev-102b7ec2-33e3-442d-8e79-72ccea10ff6a | Bonjour, nous avons tenté de vous joindre sans... | 1 | {'label': '3 stars', 'score': 0.23430046439170... |
| rev-e31e7a6c-77e7-4ad0-b7d2-b0072deea4f1 | Bonjour, nous sommes heureux d'avoir pu trouve... | 2 | {'label': '5 stars', 'score': 0.4146832823753357} |
| rev-eac63773-8559-43d0-a4ab-88fa5cc32268 | Bonjour, nous sommes heureux d'avoir pu trouve... | 2 | {'label': '5 stars', 'score': 0.4107036590576172} |
| rev-2e9eb857-a5a8-4fd4-a436-089db3782858 | Bonjour, nous avons tenté de vous joindre sans... | 2 | {'label': '3 stars', 'score': 0.23775219917297... |
| rev-4fe712bd-7451-4232-9d76-f05aaaa538c2 | Bonjour, ravi d'avoir pu vous joindre pour éch... | 1 | {'label': '1 star', 'score': 0.5053642392158508} |
| rev-c1f26089-62da-4240-8e2b-b2dd3dc37f17 | Bonjour, nous avons tenté de vous joindre sans... | 2 | {'label': '3 stars', 'score': 0.23775219917297... |
| rev-1df9979a-1ee4-4f11-a1bc-477b698a9208 | Bonjour, nous avons tenté de vous joindre sans... | 1 | {'label': '1 star', 'score': 0.23431478440761566} |
| rev-9af5a3c4-7194-441e-815d-9d211b1271f8 | Bonjour, nous avons tenté de vous joindre sans... | 2 | {'label': '3 stars', 'score': 0.22823046147823... |
| rev-9a287b65-897b-4ac8-8293-34253f09fe12 | Bonjour, nous avons essayé de vous contacter. ... | 2 | {'label': '1 star', 'score': 0.31348270177841187} |
| rev-8d6e0af5-ee33-418d-b8e3-277d46f52206 | Bonjour, nous avons tenté de vous joindre sans... | 1 | {'label': '3 stars', 'score': 0.23775219917297... |
| rev-72b1f9b8-f09d-4c59-870a-6b306eaa98b3 | Bonjour, merci d'avoir pris le temps de nous p... | 1 | {'label': '1 star', 'score': 0.4159805476665497} |
| rev-5f72a266-595f-488a-a8ff-228bf4ea6403 | Bonjour, nous avons tenté de vous joindre sans... | 2 | {'label': '3 stars', 'score': 0.23775219917297... |
| rev-e309a034-6f44-4084-837d-053563ac5882 | Bonjour, nous sommes heureux d'avoir pu trouve... | 3 | {'label': '5 stars', 'score': 0.42379361391067... |
| rev-e7d0d762-e5d4-4990-a9ee-15eb4d97e77a | Bonjour, merci d'avoir pris le temps de nous p... | 3 | {'label': '3 stars', 'score': 0.23027586936950... |
| rev-c38ab27a-e8a6-4631-91a1-5b0b92740f9f | Bonjour, nous avons tenté de vous joindre sans... | 2 | {'label': '3 stars', 'score': 0.23775219917297... |

**L'analyse de sentiment révèle que certaines réponses du service client notées faiblement par les clients (1 ou 2 étoiles) reçoivent des scores positifs (5 étoiles) de la part du modèle, suggérant un ton positif ou neutre dans la communication, peut être pour compenser la négativité du commentaire initial.** D'autres réponses sont perçues négativement (1 étoile), ce qui pourrait refléter l'admission de fautes ou le

mécontentement du client. Les réponses neutres (3 étoiles) pourraient être le résultat de l'emploi de langage standardisé ne dégageant ni positivité ni négativité.

## En Conclusion:

La plupart des réponses ont reçu un score de sentiment de '5 stars', ce qui suggère que le ton des réponses est majoritairement positif. Les réponses aux avis positifs sont formulées de manière à renforcer cette positivité.
On peut dire que les réponses fournisseurs pour les ratings de 1 à 3 sont des réponses types malgré un effort de personnalisation. Voici les justifications à cette conclusion :

Présence de Noms d'Employés : L'analyse a montré que certains noms d'employés sont fréquemment mentionnés dans les réponses. Cela indique un effort pour personnaliser les réponses, mais la récurrence de certains noms suggère que cette personnalisation peut être partie intégrante d'un modèle de réponse standardisé.

Variabilité dans l'analyse de Sentiment : Les scores de sentiment variés, allant de positif à neutre, même pour des ratings clients bas, suggèrent que les réponses sont formulées de manière à maintenir un ton constructif et professionnel. Cela peut refléter l'utilisation de phrases prédéfinies ou standardisées pour répondre aux préoccupations des clients de manière cohérente.

Répétition des Formules et Structures : La présence de phrases ou de structures répétitives, comme le montrent les analyses de contenu et de fréquence des mots, suggère que même si des noms spécifiques sont utilisés pour personnaliser, le corps principal de la réponse suit souvent un modèle préétabli.

Contraste avec les réponses aux Ratings Élevés : En comparant avec les réponses aux avis de 4 étoiles et plus, on observe une tendance similaire à utiliser des réponses formulées de manière positive. Cela renforce l'idée que, indépendamment du rating, les réponses suivent un certain script, avec des ajustements mineurs pour la personnalisation.

**Bien que les réponses aux clients avec des ratings de 1 à 3 tentent d'intégrer des éléments personnalisés, comme les noms des employés, l'ensemble des données suggère que ces réponses restent en grande partie standardisées, suivant des modèles préétablis pour maintenir un niveau de service uniforme.**

## Topic modeling:

- Objectifs : dans le but de satisfaire notre client et de **faciliter la catégorisation des commentaires clients mais aussi des réponses,** nous avons exploré le topic modeling avec le modèle Latent Dirichlet Allocation de la librairie Gensim en Python. Du fait de la standardisation des réponses fournisseurs, en effet, nous pourrions appliquer la même logique.

  Le modèle **LDA (Latent Dirichlet Allocation)** est un modèle probabiliste génératif utilisé pour le topic modeling, une technique qui permet d'identifier les sujets dominants présents dans un ensemble de documents. Il nous a aidé à identifier les sujets dominants pour chaque commentaire.

- Méthodologie: définition d'un corpus de mots - nos commentaires nettoyés et tokenisés et **rassemblés en bag of words-** sur lequel on applique le modèle et création d'un dictionnaire qui établit la correspondance entre les mots du corpus et les identifiants du modèle.

  Rappel : le bag of words permet de représenter chaque commentaire comme un ensemble non ordonné de mots et de leur fréquence respective.

  Il en ressort une distribution de probabilités qui permet de représenter des proportions et qu'il faut ensuite re-traduire en mots. **Nous avons décidé de ressortir les 4 sujets dominants de chaque commentaire.**

- Résultats **:** **L'analyse LDA nous permet de ressortir les grands thèmes liés à la vente par correspondance,** à savoir les problèmes de livraison, de paiement, de conformité de taille, etc. Elle s'avère donc relativement utile.

## Création d'un modèle interactif via une Chatbot :

- Objectifs : rendre plus conviviale et dynamique l'analyse de commentaires avec en retour, une analyse de sentiments et une liste de thèmes.

- Méthodologie : création d'une fonction dans Python qui permet de créer un user input avec comme entré un index de ligne de notre dataframe commentaires et en retour une évaluation de sentiment : de 0 à 3 pour BERT, et le classement des 4 sujets dominants dans ce commentaire en particulier.

- Résultats : **le chatbot ressort le commentaire appelé via son index et en caractères gras l'analyse de sentiments et les 4 sujets dominants**. Nous sommes ravis !

## Conclusions de l'Étape 3 :

1. -Prédiction de la note du client en fonction de son commentaire:

    a. **Feature Engineering** : Dans notre approche d'analyse des commentaires clients, **nous avons confirmé ces 3 features** : la longueur du commentaire, la longueur du titre et la durée entre la date de la transaction et le commentaire client **comme explicatives du sentiment du client**.

    b. **Class Balancing (Ré-échantillonnage) :** Cette étape de rééchantillonnage a été utilisée tout au long de nos analyses, que ce soit dans la modélisation initiale ou dans les étapes de fine tuning **pour ainsi pallier au déséquilibre très prononcé des données** ( avec 88% des commentaires sur des ratings 4 ou 5) .

    Cela a permis de s'assurer que nos modèles soient moins biaisés vers une catégorie de sentiment positif,  garantissant ainsi une évaluation plus juste de la performance.

    c. **Modélisation initiale** : nous avons appliqué une dizaine de modèles traditionnels de classification, dans un contexte classification binaire et multi-classes.

    Nous avons constaté qu'une régression logistique simple s'est avérée efficace en termes de performances avec une précision d'ensemble à 79% et une capacité à capter 67% des sentiments négatifs. Néanmoins, il a des performances très faibles en termes de précision dans la prédiction des sentiments négatifs. Près de 70% de la prédiction du modèle en sentiments négatifs sont faux.

    d. **Modélisations avancées** : Une étape préliminaire cruciale a été le nettoyage de texte pour éliminer le bruit des commentaires. Ensuite, nous avons utilisé la technique Bag of words (BoW) avec la vectorisation des commentaires nettoyés à l'aide de la fonction TF-IDF et Count Vectorizer. Dans l'ensemble des 10 modèles étudiés, nous constatons que **le modèle le plus performant avec le TF IDF s'avère être le modèle linear SVC standard avec une précision d'ensemble de 87%**, une capacité à capter 90% des sentiments négatifs et une précision de 48%.

    Avec le **Count Vectorizer, cette fois-ci, c'est la régression logistique qui obtient la meilleure performance avec 89%** de précision d'ensemble, une capacité à capter 89 % des sentiments négatifs et une précision de 51 %.

    Dans cette étape de modélisation avancée, nous avons aussi appliqué un réseau de neurones denses à 2 couches aux commentaires vectorisés via TF

IDF. Ses performances restent dans le même ordre de grandeur que les 2 modèles précédemment cités sans amélioration notable.

e. **Fine-Tuning des Modèles** : nous avons **systématiquement appliqué un Grid search CV** pour essayer d'identifier de meilleurs paramètres y compris en y intégrant TF IDF vectorizer . Malheureusement, **pas d'amélioration significative dans les performances**. Nous avons mis en œuvre des pipelines ( TF IDF > sous échantillonnage aléatoire > modèle) pour étanchéifier les données d'entraînement et de test dans le processus d'apprentissage.

   On constate que la précision des bag of words n 'est pas énorme et stagne autour des 51%, cela peut être dû au fait que le BoW ne prend pas en compte l'ordre des mots et perd le sens des commentaires. Pour pallier cela, nous avons appliqué des modèles de réseaux de neurones récurrents GRU ( Gated Recurrent Unit) capables de capturer les dépendances à long terme dans les séquences de données.

f. **Modélisation à l'aide de réseaux de neurones GRU** : **Les modèles GRU se sont avérés être nettement plus précis globalement** avec des précisions d'ensemble à plus de 91%. Ils ont aussi une précision nettement plus élevée dans la prédiction des sentiments négatifs, jusqu'à 68% pour le GRU le plus performant. Par contre, ce qu'on gagne en précision, on le perd un peu dans la capacité à capter les sentiments négatifs (recall).

   Nous avons suivi la même démarche que celle énoncée ci-dessus pour l'analyse multi classes des commentaires clients. **Il en ressort que les modèles GRU sont plus performants sur une problématique multi classes ( 3 classes de prédiction)**. Ils sont notamment plus précis dans la prédiction du sentiment négatif ou neutre. Néanmoins, ils ne captent pas aussi bien les sentiments négatifs ou neutres qu'un modèle Bag of words.

En résumé, **nous sommes impressionnés par la performance du modèle de régression logistique qui reste un modèle simple à mettre en œuvre**, que ce soit dans une problématique de classification binaire ou multi-classes. Le ré-échantillonnage a été crucial dans notre étude pour obtenir des résultats pertinents.

Quant aux modèles de deep-learning, le bon paramétrage du poids des classes ( weight-class) a permis de compenser le déséquilibre des données.

Ces étapes combinées ont permis de maximiser la qualité de notre analyse des commentaires clients et d'obtenir des résultats plus fiables et significatifs. Bien que des progrès aient été réalisés, des opportunités subsistent pour perfectionner nos modèles et affiner les paramètres des réseaux de neurones.

2. - Prédiction du retour fournisseur en fonction du commentaire client:
    a. **Similarité avec l'Analyse des Commentaires Clients** : Notre démarche pour l'analyse des commentaires fournisseurs a été parallèle à celle employée pour les commentaires clients. Nous avons adopté une approche cohérente, **en mettant l'accent sur la compréhension des verbatims** pour identifier des schémas logiques ou pas, de réponses fournisseurs.
    b. **Feature Engineering et Classification** : Tout comme dans l'analyse des commentaires clients, nous avons **investi énormément de temps dans le feature engineering, intégrant des techniques telles que les word embeddings, les n-grams pour extraire des informations plus riches.** La classification des commentaires fournisseurs a été effectuée en utilisant des modèles adaptés au langage naturel.
    c. **Recherche d'une trame Standard** : Un aspect spécifique de notre analyse des commentaires fournisseurs a été la recherche d'une trame standard en fonction des notes attribuées par les clients. Nous avons examiné si certains commentaires spécifiques ou thèmes récurrents étaient associés à des évaluations particulières. Cela nous a permis d'**identifier des sentiments et des tendances et des structures génériques des réponses fournisseurs**.
    d. **Modélisation et Fine-Tuning** : le fine-tuning a été réalisé à travers des itérations successives, en utilisant des modèles non supervisés tels que Vader, Bert, K-means qui ont prouvé leurs limites en travaillant sur du multi-classes et probablement dû au fait que nous avons beaucoup de réponses génériques sur la partie fournisseurs.

    En conclusion, notre approche pour l'analyse des commentaires fournisseurs a été alignée sur celle des commentaires clients, assurant une cohérence dans notre méthodologie. **Les limites atteintes par les modèles nous on fait rechercher de nouveaux axes de travail tels que l'analyse des verbatims. Ce travail nous a amené à confirmer le caractère générique des réponses fournisseurs** pour essayer de les standardiser. Cette partie restera à développer.

3. - Analyse de sentiments client/fournisseur:

    On constate à travers nos diverses itérations que **l'analyse de texte peut s'avérer ardue et reste un processus très itératif.** Il en ressort aussi que certains outils vont très bien identifier la neutralité d'un commentaire mais par contre ne vont pas supporter une bonne prédiction de note. En résumé, voila ce qui ressort de notre étude

    <u>Approches non supervisées :</u>

a. **Word Cloud** : En commençant par créer des "Word Cloud" à partir des commentaires des clients, nous avons rapidement accédé à un outil de visualisation qui a aussi permis de mettre en évidence les mots les plus fréquemment utilisés, offrant ainsi une vue rapide des thèmes dominants et des préoccupations générales. La limitation des word clouds étant vraiment dans l'utilisation des stop words et l'affinage des mots à enlever à chaque itération. Sinon, **il s'agit d'un outil très pratique, qui combiné à d'autres techniques comme la vectorisation peut se révéler intéressant.**

b. **Sentiment Analysis** : En utilisant l'analyse de sentiments, nous avons évalué la tonalité émotionnelle des commentaires. Cela nous a permis de classer les retours des clients en catégories telles que positif, négatif ou neutre, offrant ainsi un aperçu global du sentiment associé aux expériences des clients. **Nous avons constaté beaucoup de neutralité qui s'explique car les commentaires ne sont pas que négatifs et mettent aussi l'accent sur l'expérience positive avec le site de la Redoute.**

c. **BERT** (Bidirectional Encoder Representations from Transformers) : Afin d'obtenir une compréhension plus contextuelle et nuancée, nous avons utilisé BERT, qui s'est avéré plus proche du rating client que ce que Vader avait pu nous fournir, **améliorant ainsi la précision de l'évaluation du sentiment.**

Approches semi supervisées ou supervisées :

d. **Naïve Bayes** : Pour une analyse plus approfondie, nous avons utilisé un modèle de classification Naïve Bayes**. Ce modèle statistique a été formé sur des commentaires annotés manuellement** pour prédire automatiquement la catégorie de sentiment des commentaires non étiquetés. La constatation ici est **qu'il aurait fallu beaucoup plus de supervision du modèle pour ressortir des résultats probants.**

e. **Topic Modeling** : Enfin, pour découvrir les sujets sous-jacents et classifier ces commentaires, nous nous sommes intéressés au "Topic Modeling". Cela a permis d'identifier les thèmes récurrents et les sujets spécifiques abordés par les clients mais aussi par le fournisseur, offrant ainsi une **bonne vue d'ensemble des préoccupations et des réponses clés.** Cette fonctionnalité a été appliquée avec du LDA.

En combinant ces différentes étapes, nous avons pu obtenir une **meilleure compréhension des commentaires des clients**, allant au-delà de simples tendances de mots. Ces approches successives ont **permis de cerner les sentiments, de classifier les commentaires, d'analyser le contexte avec BERT et de découvrir les thèmes spécifiques grâce au topic modeling.**

Création d'un modèle interactif via une Chatbot : nous avons finalisé notre projet en travaillant sur un modèle de chatbot simple à des vues de démonstration.

- Objectifs : rendre plus conviviale et dynamique l'analyse de commentaires avec en retour, une analyse de sentiments et du topic modeling .

- Méthodologie : création d'une fonction dans Python qui permet de créer un user input avec comme entré un index de ligne de notre dataframe commentaires et en retour une évaluation de sentiment( de 0 à 3 pour BERT) et le classement des 4 sujets dominants dans ce commentaire en particulier.

- Résultats : le chatbot ressort le commentaire appelé via son index et en caractères gras l'analyse de sentiments et les 4 sujets dominants. Nous sommes ravis !

# Annexes

## Rapport d'exploration des données :

Le rapport d'exploration des données des disponible dans le GitHub projet via le lien suivant : "02_edf_rapport_'exploration_données_v1.0.xlsx"

## Notebooks du projets :

Les notebook jupyters du projet sont disponibles dans le GitHub Projet :
Jupyter notebooks des programmes disponibles dans le Git projet

***Modèle baseline*** :
01_baselines_models_04.ipynb
01_baselines_multiclass_02.ipynb

***Modèles Bag of Words***
02_text_processing_0.0.ipynb
02_vectorization_clf_multiclass_0.1.ipynb
02_vectorization_cvz_clf_binary_0.4.ipynb
02_vectorization_nn_models_02.ipynb
02_vectorization_tfidf_clf_binary_0.4.ipynb
02_vectorization_tfidf_gridscv_clf_binary_0.4.ipynb

***Réseaux de neurones récurrents GRU***
03_Sentiment_analysis_with_RNN_04.ipynb
03_Text_Processing_for_RNN.ipynb

***Analyse de sentiments TexBlob, Vader, Bert***
04_Sentiment_Analysis_tb_vd_0.2.ipynb
04_Sentiment_analysis_BERT_0.0.ipynb
02_mib_eda_redoute_suppliers_Analysis_Sentiment_VADER_ROBERTA
02_mib_eda_redoute_suppliers_Diagnostic_Réponses_Rating_Supérieur_à_4
02_mib_eda_redoute_suppliers_Diagnostic_Réponses_Rating_Inferieur_ou_egale_a_3

***Topic Modeling***

05_topic_modeling_de_BERTopic.ipynb
05_topic_modeling_gensim.0.0.ipynb
Essai_Classif_personnalisée: Naive Bayes et pré-entraînement manuel du modèle
Essai_Chatbot : Bert, LDA et Chatbot

**DataViz**

06_dviz_models_performance_02.ipynb