**1. Prétraitement des données**

Un prétraitement cohérent et rigoureux est une étape essentielle pour améliorer la performance des modèles de classification, en particulier dans un contexte de tweets, caractérisé par une forte variabilité linguistique, une syntaxe bruitée, et un vocabulaire très contextuel. Nous avons adopté une stratégie de prétraitement adaptée à chacune des trois tâches, tout en maintenant une cohérence globale suivant cette ordre de taches :

-Nettoyage textuel et des stop words,

-Normalisation linguistique : Lemmatisation et tokenisation.

-Vectorisation avec TF-IDF et paramétrage des n-grammes.

-Gestion du déséquilibre.

**NOTE** : Des tests ont été effectués dans le cadre de cette classification en utilisant uniquement les attributs importants extraits à l’aide du classifieur Random Forest. Cependant, les résultats n’étaient pas très satisfaisants. Nous avons donc envisagé de réduire manuellement la taille maximale des features lors de la vectorisation, afin d’observer le comportement du modèle. Cette décision se justifie par la présence de bruit dans les données, bruit qui reste néanmoins représentatif pour distinguer les tweets scientifiques des non-scientifiques. C’est pourquoi nous avons choisi de ne pas effectuer une classification uniquement à partir de ces attributs.

Ces étapes ont été adaptées à chaque type de classification ({SCI} vs. {NON-SCI}, {CLAIM, REF} vs. {CONTEXT}, {CLAIM} vs. {REF} vs. {CONTEXT}) afin d’obtenir les meilleures performances. Dans la suite de ce rapport, une comparaison entre les données brutes et les données après traitement sera présentée, accompagnée d’une justification des choix effectués.

### Partie 1 — Classification SCI vs NON-SCI ( classification binaire)

Cette tâche consiste à distinguer les tweets en lien avec la science (SCI) de ceux qui ne le sont pas (NON-SCI). La abse de donnée contient donc des tweets très hétérogènes, ce qui justifie un prétraitement rigoureux mais généraliste.

* Nous avons supprimé les mentions (@), les caractères spéciaux et converti le texte en minuscules. Cela permet de standardiser l’entrée pour éviter que des variations superficielles n’influencent la classification.
* Nous avons utilisé la liste anglaise de NLTK enrichie de mots fréquents sur Twitter ("http", "https", "rt", "co", "amp", "via"), car ces éléments sont omniprésents mais peu informatifs pour déterminer la nature scientifique du contenu.
* **Vectorisation TF-IDF (1-2-grammes)** : un choix raisonnable pour capturer à la fois des mots individuels et des associations fréquentes, tout en limitant la complexité.
* **SMOTE** : utilisé pour équilibrer la distribution entre classes SCI et NON-SCI, car ces dernières sont très déséquilibrées (moins de 10 % de SCI).

Les figures suivantes montrent la distribution des classes (SCI vs NON-SCI), ainsi que la représentation des nuages de mots et les diagrammes en barres sur lesquels nous nous sommes appuyés pour éliminer les stop words et les mentions (@).

📊 **Figures à inclure** :

* Histogramme nbr tweet scientifique et non scientifique
* Nuage de mots et barplot des mots les plus fréquents (pour les @ et cette liste : "https", "rt", "co", "amp", "via"
* Distribution des classes (SCI vs NON-SCI)

**une phrase d’analyse des figures selon le resultats,**

### 🔹 Partie 2 — Classification CLAIM\_OR\_REF vs CONTEXT (binaire, SCI uniquement)

Cette tâche vise à distinguer deux types d’usages des tweets scientifiques : ceux qui formulent ou citent une affirmation scientifique (CLAIM/REF) vs ceux qui servent de contexte (CONTEXT). Les tweets analysés sont uniquement ceux classés comme SCI dans la première tâche.

* Seuls les tweets marqués comme scientifiques (science\_related = 1) ont été conservés.
* **Prétraitement du texte** : très similaire à la première tâche (minuscules, stop words, lemmatisation), mais **sans suppression des mentions et caractères spéciaux**, afin d’explorer s’ils peuvent porter une valeur contextuelle utile.
* **Vectorisation TF-IDF (1-3-grammes)** : on utilise ici des trigrammes pour mieux capturer des constructions typiques d'énoncés de type affirmation ou citation.
* **Pas de SMOTE classique** : Ici, le déséquilibre est partiel (tableau numeo,,) mais la classification est multi-label (un tweet peut être à la fois CONTEXT et CLAIM/REF). Nous avons donc implémenté un rééchantillonnage manuel spécifique au multi-label, en nous basant sur l’identification de toutes les combinaisons de labels possibles et en suréchantillonnant celles qui étaient sous-représentées.

tableau : Répartition des tweets selon les combinaisons des labels multi-label (CLAIM/REF et CONTEXT)

| is\_claim\_or\_ref | is\_context | Nombre de tweets |
| --- | --- | --- |
| 1 | 1 | 218 |
| 1 | 0 | 124 |
| 0 | 1 | 33 |

en évidence un déséquilibre partiel dans la distribution des combinaisons de labels multi-label. La combinaison {11} est majoritaire, tandis que la combinaison {01} est fortement sous-représentée. Étant donné la nature multi-label du problème (un tweet pouvant appartenir à plusieurs classes), un rééquilibrage des données s’avère nécessaire pour éviter que le modèle ne soit biaisé en faveur des classes majoritaires. Pour cela, un suréchantillonnage ciblé des combinaisons rares a été mis en place.

📊 **Figures à inclure** :

* nuage des points par class
* barplot des mots les plus fréquents (pour les @ et cette liste : "https", "rt", "co", "amp", "via"
* Distribution des classes (SCI vs NON-SCI)

qlq phrase de l’analyse des figures ou legends ,

### 🔹 Partie 3 — Classification CLAIM vs REF vs CONTEXT (multi-classes, SCI uniquement)

Cette tâche est la plus fine : chaque tweet scientifique est classé comme revendication (CLAIM), référence (REF) ou contexte (CONTEXT). C’est une classification à 3 classes, mais les tweets peuvent avoir plusieurs labels (multi-label).

* Seuls les tweets marqués comme scientifiques (science\_related = 1) ont été conservés.
* Nettoyage classique, mais **avec une étape supplémentaire** : remplacement des URLs par le token "URL", afin de conserver leur présence (utile pour REF) sans garder l’URL brute car la classe REF étais la moine bonne prédite
* **Stop words** : liste standard + mots Twitter ("rt", "co", "amp", "via"); pas de suppression de négation ici, car certaines peuvent indiquer une nuance importante.
* **Vectorisation TF-IDF (1-3-grammes)** : les trigrammes sont particulièrement utiles pour différencier entre REF (souvent formalisé), CLAIM (langage assertif) et CONTEXT (vocabulaire plus descriptif).
* **Suréchantillonnage manuel multi-label** : comme les classes sont très déséquilibrées (par exemple, très peu de REF), on a implémenté un suréchantillonnage par combinaison de labels, sans détruire la nature multi-étiquette.

**📊Figures à inclure** :

* nuage des points [CLAIM, REF, CONTEXT]
* Histogramme de fréquence des classes
* barplot frequnec des url + "rt", "co", "amp", "via"

soit legend de chaque figure ou une analyse generale des figure ca depond.

**2. Évaluation comparative des performances de classification**

Dans cette section, nous comparons les performances des modèles sur trois tâches de classification hiérarchiques successives. Ces tâches sont les suivantes :

* **Tâche 1 : classification binaire : Distinguer les tweets scientifiques des non-scientifiques ({SCI} vs. {NON-SCI}),**
* **Tâche 2** : **Classification Binaire Multi-Label:** Différencier les tweets scientifiques portant des affirmations ou des références de ceux apportant un simple contexte ({CLAIM, REF} vs. {CONTEXT}).
* **Tâche 3** :**classification multi-classe et multi-label non exclusive :**  Classifier plus finement les tweets scientifiques en trois catégories ({CLAIM} vs. {REF} vs. {CONTEXT}) .

Dans la suite, nous présentons les résultats de classification obtenus pour chaque tâche ainsi qu’une comparaison des performances des approches utilisées.

**2.1 **Tâche 1 : classification binaire ({SCI} vs. {NON-SCI})****

**Dans cette première tâche, nous avons comparé plusieurs modèles de classification pour distinguer les tweets scientifiques des non-scientifiques.L'objectif était d'évaluer les performances de différentes approches d'apprentissage supervisé appliquées à un problème binaire.les chois suivant on etait effectué :**

* **Modèles évalués** : Régression logistique, Naive Bayes, k-Nearest Neighbors (k-NN), Random Forest, Support Vector Machine (SVM).
* **Optimisation des hyperparamètres** :  
  Chaque modèle a été optimisé à l’aide d’une **recherche par grille (GridSearchCV)** afin d’identifier les meilleures combinaisons d’hyperparamètres.
* **Évaluation des performances** par une **validation croisée avec KFold (k = 10)** ce qui permet une estimation robuste des performances sur différentes partitions du jeu de données.

Le tableau suivant résume les résultats obtenus pour chaque modèle en termes de **précision**, **rappel**, **F1-score** et **exactitude (accuracy)**, calculés à la fois sur les validations croisées et sur le jeu de test.

**Tableau 1 – Comparaison des performances des classifieurs après optimisation des hyperparamètres (10-fold cross-validation et évaluation sur le jeu de test)**

| Modèle | Accuracy (moyenne ± std) | Precision (macro) | Recall (macro) | F1-score (macro) |
| --- | --- | --- | --- | --- |
| **Logistic Regression** | 0.9327 ± 0.0203 | 0.93 | 0.93 | 0.93 |
| **Naive Bayes** | 0.9092 ± 0.0218 | 0.91 | 0.90 | 0.90 |
| **k-NN** | 0.8092 ± 0.0337 | 0.83 | 0.80 | 0.80 |
| **Random Forest** | 0.8680 ± 0.0235 | 0.86 | 0.84 | 0.83 |
| **SVM** | 0.9412 ± 0.0185 | 0.94 | 0.94 | 0.94 |

**Le modèle SVM semble être le plus performant pour cette tâche, avec les meilleurs scores en précision, rappel et F1-score, ainsi qu'une exactitude (accuracy) accompagnée du plus faible écart-type, tant sur le jeu de test que lors de la validation croisée comme le montre le boxplot de la figure (,,,). Cependant, la régression logistique reste un modèle très solide et constitue une alternative intéressante, surtout pour les configurations où l'on cherche un compromis entre précision et rappel.**

**Afficher ici la matrci de confusion aussi et dir le nombre de tweet mal predit comme commentrair**

**box plot de raphaelle**

La **validation croisée avec KFold** a permis d'obtenir une estimation robuste de la performance des modèles, garantissant ainsi que les résultats observés ne sont pas biaisés par une partition particulière des données. Cela rend les résultats de cette analyse comparativement plus fiables et adaptés pour une généralisation à de nouvelles données.

**2.2 **Tâche 2 : Classification Binaire Multi-Label ({CLAIM, REF} vs. {CONTEXT})****

**Dans cette deuxième tâche, nous avons comparé plusieurs modèles de classification pour résoudre un problème de classification multi-label, où chaque tweet peut être associé à plusieurs catégories simultanément. L’objectif était d’évaluer les performances des différentes approches d’apprentissage supervisé adaptées à un contexte multi-label. Les choix suivants ont été effectués :**

• **Modèles évalués** : Régression logistique, Naive Bayes, k-Nearest Neighbors (k-NN), Random Forest, Support Vector Machine (SVM), Gradient Boosting.

• **Optimisation des hyperparamètres** : Chaque modèle a été optimisé à l’aide d’une recherche par grille (GridSearchCV) pour identifier les meilleures combinaisons d’hyperparamètres.

• **Évaluation des performances par validation croisée avec KFold (k = 10)**

Le tableau suivant présente les résultats obtenus pour chaque modèle, en termes de précision, rappel, F1-score et exactitude (accuracy), calculés à la fois sur les validations croisées et sur le jeu de test.

**Tableau 1 – Comparaison des performances des classifieurs après optimisation des hyperparamètres (10-fold cross-validation et évaluation sur le jeu de test)**

| Modèle | Accuracy (moyenne ± std) | Precision (macro) | Recall (macro) | F1-score (macro) |
| --- | --- | --- | --- | --- |
| **SVM (RBF)** | 0.9183 ± 0.0193 | 0.99 | 1.00 | 0.99 |
| **Naive Bayes** | 0.8647 ± 0.0292 | 0.93 | 0.93 | 0.93 |
| **k-NN** | 0.8876 ± 0.0183 | 0.99 | 0.99 | 0.99 |
| **Random Forest** | 0.9359 ± 0.0162 | 0.99 | 1.00 | 0.99 |
| **Logistic Regression** | 0.8961 ± 0.0232 | 0.98 | 0.97 | 0.97 |
| **Gradient Boosting** | 0.9236 ± 0.0162 | 0.99 | 1.00 | 0.99 |

**Les modèles SVM et Random Forest ont montré des performances globales exceptionnelles, particulièrement dans le cadre d'un problème de classification multi-label comme celui-ci.** L'optimisation des hyperparamètres via GridSearchCV a permis d'améliorer significativement les performances de chaque modèle en ajustant les paramètres clés.

Par contre le k-NN a montré des résultats solides avec des scores de précision et de rappel très élevés (0.99). Cependant, il a un léger sur-apprentissage, ce qui peut être un facteur limitant pour de très grands ensembles de données.

****-boxplot plus matrcice de confusion de svm****

****2.3 Tâche 3 :** classification multi-classe et multi-label non exclusive **({CLAIM} vs. {REF} vs. {CONTEXT})****

| ****Modèle**** | ****Accuracy moyenne**** ± écart-type | ****Hamming Loss**** ± écart-type | ****F1-Score**** ± écart-type | ****Rappel**** ± écart-type |
| --- | --- | --- | --- | --- |
| **LinearSVC** | 0.7171 ± 0.0625 | 0.2844 ± 0.0625 | 0.8844 ± 0.0625 | 0.75 ± 0.04 |
| **Random Forest** | 0.7836 ± 0.0507 | 0.2489 ± 0.0507 | 0.9133 ± 0.0507 | 0.83 ± 0.05 |
| **SVM** | 0.7331 ± 0.0691 | 0.2622 ± 0.0691 | 0.8992 ± 0.0691 | 0.89 ± 0.07 |
| **Naive Bayes** | 0.6101 ± 0.0802 | 0.3156 ± 0.0802 | 0.8601 ± 0.0802 | 0.82 ± 0.07 |
| **k-NN** | 0.6868 ± 0.0763 | 0.36 ± 0.0763 | 0.8615 ± 0.0763 | 0.77 ± 0.06 |
| **Logistic Regression** | 0.7433 ± 0.0545 | 0.2578 ± 0.0545 | 0.8933 ± 0.0545 | 0.79 ± 0.06 |
| **Gradient Boosting** | 0.7696 ± 0.0476 | 0.2356 ± 0.0476 | 0.9086 ± 0.0476 | 0.85 ± 0.05 |

**Dans ce tableau, on observe que tous les modèles de classification présentent une précision et un F1-score relativement bons, avec des valeurs de Hamming Loss plus faibles pour des modèles comme Random Forest, Gradient Boosting et Logistic Regression, ce qui indique une capacité à mieux classer les données tout en minimisant les erreurs globales. La précision est généralement élevée pour les classes claim et context, mais elle est un peu plus faible pour la classe reference, ce qui peut expliquer pourquoi ces modèles ont des performances variées. Cela montre que les modèles sont bons pour identifier les assertions et les contextes, mais ont plus de difficulté à classer correctement les références, ce qui peut être dû à une classe reference plus difficile à différencier ou moins représentée dans le jeu de données.**

Le **Hamming Loss** est relativement faible dans la plupart des modèles, ce qui signifie que, bien que des erreurs puissent se produire, ces erreurs ne sont pas trop fréquentes, ce qui est un bon signe pour la qualité des prédictions des modèles. Cependant, le **F1-score** est parfois un peu plus élevé que le score d'accuracy, indiquant que les modèles cherchent à optimiser un compromis entre précision et rappel pour chaque classe.

La **Hamming Loss** plus élevée pour des modèles comme **k-NN** et **Naive Bayes** est probablement le reflet de leur difficulté à bien distinguer toutes les classes, particulièrement pour la classe **reference**, qui semble poser plus de défis en termes de précision et de rappel. Cela explique aussi pourquoi ces modèles, bien que performants dans certaines classes, peuvent avoir des scores globaux de précision et de rappel moins élevés par rapport à d'autres, comme **Random Forest** ou **Gradient Boosting**, qui semblent mieux gérer la classification dans l'ensemble.