Machine Learning I

Classification d'assertions venant de Twitter selon leur rapport à la science.

Auteur :

AGÜERA SANCHEZ Martina (22101917)

BOUBY Lucien (22011624)

GONG Ni (22025419)

MACKOW Anaïs (22016204)

Master 1 Informatique - Parcours IASD

2024 - 2025

Table des matières

1 Introduction **3**

2 Analyse des données **3**

3 Classifications **4**
3.1 SCI vs. NON-SCI 5
3.2 CLAIM, REF vs. CONTEXT 5
3.3 CLAIM vs. REF vs. CONTEXT 6
3.3.1 Stratégie MultiOutput multi-classe 6
3.3.2 Stratégie One vs One 7
3.3.3 Stratégie One vs Rest 7
3.3.4 stratégie Classifieur Chaîné multi-classe 8
3.3.5 stratégie LSTM multi-label 8

4 Analyses et Conclusions **9**

5 Ouverture **10**

6 Annexe **11**

7 Bibliographie **12**

1 Introduction

Le but de ce projet de Machine Learning est de réaliser une classification d'assertions venant du réseau social X (anciennement Twitter) selon leur rapport à la science. Pour cela nous utiliserons la base de données SciTweets. Ces données sont catégorisées de plusieurs façons. Les tweets sont classés par rapport à un label binaire: "science related" parmi les tweets scientifiques on retrouve trois sous catégories : "claim/question" , "reference" et "research context". Sachant qu'un tweet peut appartenir à aucune, à une ou à plusieurs classes.

2 Analyse des données

Premièrement, il est essentiel de procéder à une phase d'exploration des données. Pour ce faire, nous avons examiné plusieurs tweets issus du dataset base afin d'en comprendre la structure textuelle. Cette première inspection révèle que les tweets sont souvent rédigés de manière informelle, comportant de nombreux caractères spéciaux (tels que , @, liens URL, emojis, etc.) ainsi que des abréviations.

Afin d'obtenir une vue d'ensemble du corpus, qui contient près de 1 200 tweets, nous avons généré un nuage de mots mettant en évidence les mots les plus fréquents du corpus . Sans surprise, les termes les plus représentés sont des mots-outils (tels que "the" , "a" ou encore "and"), généralement considérés comme des stop words. Ces mots, n'apportant que peu d'information sémantique, seront supprimés lors des étapes de prétraitement.

En complément du word cloud, nous avons réalisé une analyse des bi/tri-gramme. Par exemple, cela nous permet de vérifier si l'expression "climate change" apparaît fréquemment dans les tweets classés comme *science related*. Cette analyse nous a notamment permis de constater que certaines classes partagent des thématiques communes : le trigramme le plus fréquent dans les classes *scientific reference* et *scientific context* est par exemple "molecular map animal".

Nous avons également examiné la taille du jeu de données et la répartition des labels jugées insuffisantes. Nous avons donc procédé à une augmentation des données via ChatGPT, en veillant à ne pas altérer la cohérence du corpus. Pour cela, nous avons comparé les corrélations entre les classes *Claim*, *Reference* et *Context* avant et après augmentation. Cette démarche nous a permis d'améliorer la distribution pour la classification binaire, ainsi que d'augmenter le nombre d'occurrences des sous-catégories scientifiques. Le tableau ci-dessous présente un comparatif des répartitions. Les pourcentages pour les deux dernières classifications sont rapportés au total des tweets *science-related*.

Table 1: Comparaison des distributions avant et après data augmentation

Tâche	**Classe**	**Nb initial**	**% initial**	**Nb après aug.**	**% après aug.**
SCI vs. NON-SCI	SCI	765	67.1%	2 393	52.4%
	NON-SCI	375	32.9%	2 170	47.6%
CLAIM+REF vs. CONTEXT	CLAIM + REF	342	44.7%	1 737	72.6%
	CONTEXT	251	32.8%	1 448	60.5%
CLAIM vs. REF vs. CONTEXT	CLAIM	263	34.3%	1 160	48.4%
	REF	203	26.5%	969	40.5%
	CONTEXT	251	32.8%	1 448	60.5%

Comparaison	Dataset augmenté	Dataset original
Claim vs. Reference	0.42	0.18
Claim vs. Context	0.78	0.61
Reference vs. Context	0.66	0.55

Table 2: Corrélation entre les différentes classes dans le dataset augmenté et le dataset original.

Les corrélations dans le dataset augmenté sont globalement plus élevées que celles observées dans le dataset original. En particulier, la corrélation entre *Claim* et *Context* est nettement plus forte dans le dataset augmenté. Cela suggère que l'augmentation des données a permis de renforcer les liens entre les classes, tout en conservant des relations cohérentes et logiques. Ainsi, bien que les données aient été augmentées, elles respectent les tendances principales du dataset initial, ce qui garantit que l'augmentation n'a pas altéré l'intégrité des relations entre les classes.

Plusieurs prétraitements ont été appliqués, incluant l'expansion des abréviations, la suppression des emojis et caractères spéciaux, le remplacement des années et URL par des tokens, la mise en minuscules et la lemmatisation. Cette dernière permet de réduire la variabilité lexicale, rendant les textes plus homogènes. Des alternatives comme la stemmatisation ou la normalisation ont été testées, sans amélioration notable sur les performances.

Il a également été nécessaire de convertir les textes en représentations numériques exploitables. Nous avons opté pour la pondération TF-IDF, bien adaptée aux textes courts comme les tweets, car elle met en valeur les mots porteurs de sens en équilibrant fréquence locale et rareté globale dans le corpus.

Afin d'évaluer la structure de nos données textuelles, nous avons utilisé l'algorithme *t-SNE* (*t-distributed Stochastic Neighbor Embedding*) afin de projeter des vecteurs (issus par exemple de TF-IDF ou d'embeddings)

Cette visualisation nous permet donc d'observer la répartition des tweets dans l'espace vectoriel, d'identifier visuellement d'éventuels regroupements ou recouvrements entre classes et d'évaluer la pertinence de notre encodage textuel pour la tâche de classification. Ci-joint dans l'annexe les Word-Clouds et t-SNEs (cf.Annexe) nous ayant aider de visualiser nos données et l'impact des prétraitements.

3 Classifications

Pour évaluer nos classifications, nous nous sommes appuyés sur le F1-score, l'écart-type des performances (illustré en annexe par des boîtes à moustaches) et l'équilibre des matrices de confusion. Bien que non incluses dans ce document par souci d'espace, celles-ci ont été prises en compte à chaque étape pour guider nos choix.

Nous avons utilisé l'outil *Optuna* pour optimiser les hyperparamètres des modèles, avec une validation croisée interne pour sélectionner les meilleurs paramètres, et une validation croisée externe pour évaluer leur robustesse.

Enfin, afin de comparer les approches traditionnelles à des modèles profonds, nous avons testé un réseau *LSTM* (Long Short-Term Memory) avec une sortie sigmoïde, bien adapté à la classification binaire. Ce type de modèle tient compte de l'ordre des mots, ce qui est particulièrement pertinent pour des textes courts comme les tweets.

3.1 SCI vs. NON-SCI

Dans cette première classification le but est de séparer les tweets scientifiques des non scientifiques. Il s'agit donc d'une classification binaire. Sur cette première classification on ne prend pas en compte les sous catégories de l'attribut *science related*.

Models	**Scitweets**	**Dataset augmenté**		**Dataset downsamplé**	
	Eval/Test	**Eval/Test**	**Ext***	**Eval/Test**	**Ext***
Naïves Bayes	0.56	0.71	0.58	0.73	0.42
Logistic Regression	0.70	0.77	0.54	0.80	0.38
Decision Tree	0.67	0.76	0.47	0.77	0.23
KNeighbours	0.55	0.79	0.53	0.80	0.45
Random Forest	0.71	0.76	0.56	0.81	0.37
SVC	0.67	0.77	0.53	0.80	0.35
Xgboost	0.62	0.81	0.55	0.81	0.43
LSTM	0.40	0.55	0.54	0.55	0.17

Table 3: Première Classification : Performance des modèles sur différents jeux de données.

Les pourcentages présentés correspondent au F1-score, défini comme la moyenne harmonique entre la précision (precision) et le rappel (recall). Ainsi il constitue un indicateur pertinent dans le contexte de classes déséquilibrées.

L'annotation *Ext** fait référence à la validation externe, où l'ensemble des tweets du jeu de données initial est prédit par les modèles optimisés.

Malgré des performances globales modestes, les résultats ne révèlent pas de biais significatif en faveur d'une classe. Par ailleurs, le meilleur modèle entraîné sur le jeu de données initial - un classifieur XGBoost - atteint un F1-score de 0.85 en validation externe sur ce même jeu, ce qui suggère une bonne généralisation et l'absence de sur-apprentissage.

Enfin, nous avons procédé à un downsampling du jeu de données augmenté, afin de mieux équilibrer les classes. Cette étape a été motivée par la sur-représentation de certaines classes après augmentation, et visait à renforcer la robustesse des modèles tout en limitant les déséquilibres structurels.

3.2 CLAIM, REF vs. CONTEXT

Nos analyses textuelles nous ont révèlé une forte similarité entre les classes *CLAIM/REF* et *CONTEXT*, notamment entre les sous-classes *Reference* et *Context*, qui partagent des termes fréquents comme "research" ou "scientist".

Pour corriger le déséquilibre entre les classes, nous avons dupliqué certaines instances des classes *Reference* et *Context*, en considérant *Reference* comme équivalente à un label nul. Leur forte corrélation initiale permettait cette opération sans compromettre la cohérence des données. L'objectif était d'augmenter le volume d'exemples dans les classes minoritaires tout en préservant la répartition globale, afin de renforcer l'apprentissage sans déformer la frontière de décision, principalement guidée par la classe *Claim*.

Models	Scitweets	Dataset augmenté		Dataset downsamplé	
		Eval/Test	Ext*	Eval/Test	Ext*
Naïves Bayes	0.61	0.55	0.64	0.72	0.58
Logistic Regression	0.68	0.65	0.60	0.75	0.49
Decision Tree	0.65	0.63	0.49	0.75	0.42
KNeighbours	0.57	0.64	0.56	0.77	0.57
Random Forest	0.67	0.66	0.52	0.78	0.51
SVC	0.67	0.65	0.64	0.73	0.54
Xgboost	0.63	0.61	0.61	0.77	0.50
LSTM	0.62	0.75	0.28	X	X
Embedded	0.10	0.13	X	X	X

Table 4: Deuxième Classification : Performance des modèles sur différents jeux de données.

Embedded Dans cette approche, nous procédons d'abord à une classification *Claim* vs. *Reference/Context*, puis, si un tweet est classé comme négatif, nous effectuons une classification *Reference* vs. *Context*. Si le tweet est classé comme appartenant à la classe *Claim/Reference*, il est marqué comme tel ; sinon, il est classé comme *Context*.
Pour chacun des deux modèles de classification, une recherche hyperparamétrique via Optuna est effectuée pour trouver les meilleurs paramètres. Comme montré ci-dessus, les résultats n'ont pas été ceux escomptés.

3.3 CLAIM vs. REF vs. CONTEXT

Cette classification multi-label s'est révélée particulièrement intéressante, car elle permet de reformuler le problème en classification multi-classe via l'approche *Label Powerset*. Celle-ci consiste à encoder chaque combinaison de labels (par exemple `000`, `001`, `011`, etc.) comme une classe distincte. Ainsi, bien qu'un tweet puisse appartenir à plusieurs catégories, cette transformation rend le problème compatible avec des classifieurs multi-classes classiques.

3.3.1 Stratégie MultiOutput multi-classe

Classe	F1-score	Support
Context	0.22	7
Ref/Context	0.43	14
Claim	0.69	27
Claim/Context	0.00	2
Claim/Ref/Context	0.52	25
Accuracy	0.53 (75)	
Macro Moyenne	0.37	
Moyenne Pondérée	0.52	

Table 5: XGBoost – dataset Scitweets

Classe	F1-score	Support
Context	0.30	86
Ref/Context	0.52	128
Claim	0.70	133
Claim/Context	0.42	11
Claim/Ref/Context	0.45	76
Exactitude	0.51 (434)	
Macro Moyenne	0.48	
Moyenne Pondérée	0.52	

Table 6: XGBoost – dataset augmenté

On observe que la classe *Claim* est celle qui obtient les meilleures performances. Cela peut s'expliquer par une plus faible corrélation avec les autres classes, ce qui la rend plus facilement identifiable. À l'inverse, les classes combinées comme *Claim/Context* ou *Ref/Context*, plus ambiguës, obtiennent des scores nettement inférieurs.

3.3.2 Stratégie One vs One

La stratégie *One-vs-One* consiste à transformer un problème multi-classes en une série de classifications binaires, en entraînant un classifieur pour chaque paire de classes. Pour n classes, cela implique l'entraînement de $\frac{n(n-1)}{2}$ modèles. Lors de la prédiction, chaque modèle vote pour une classe, et l'instance est assignée à celle ayant obtenu le plus de votes. Cette approche permet de mieux exploiter les distinctions locales entre classes, mais devient rapidement coûteuse en calcul lorsque le nombre de classes est élevé.

Models	**Dataset augmenté**		**Scitweets**
	Eval/Test	**Ext***	
Random Forest	0.47	0.48	0.36
SVC	0.40	0.47	0.30
Naïves Bayes	0.42	0.43	0.28
KNeighbours	0.45	0.48	0.29
Decision Tree	0.40	0.44	0.31
Logistic Regression	0.40	0.45	0.32
Xgboost	0.46	0.46	0.33

Table 7: Troisième classification : F1-score des modèles appris sur le jeu d'apprentissage et sur les tweets scientifique du jeu de données d'origine avec la méthode OvO.

Les résultats ne sont pas très bons. Ceci s'explique par le fait que nous avons augmenter le nombre de classes à reconnaître sans pour autant augmenter la quantité de données dans le dataset. Chaque nouvelle classe contient donc moins de ligne que les classes précédentes, les estimateurs ont donc plus de mal à les différencier. De plus certaines classes sont largement sous-représentée. Nous n'avons pas généré de nouvelles données pour celles-ci car elles étaient déjà très peu représenté dans le dataset fourni. Ainsi, si nous générions 100 lignes à partir de 4 nous aurions perdu toute cohérence avec les données d'origines.

3.3.3 Stratégie One vs Rest

Une deuxième solution est de rajouter les combinaisons de labels multi-label en nouvelles classes uniques (ex. Claim ET Context, Reference ET Context ...). De cette manière chaque classe est comparée aux restantes. La classification reste donc non-binaire car elle a encore plus de classes différentes mais n'est plus multilabel.

Models	**Dataset augmenté**		**Scitweets**
	Eval/Test	**Ext***	
Random Forest	0.61	0.60	0.47
SVC	0.57	0.58	0.54
Naïves Bayes	0.41	0.42	0.18
KNeighbours	0.56	0.53	0.31
Decision Tree	0.54	0.59	0.55
Logistic Regression	0.57	0.59	0.54
Xgboost	0.59	0.56	0.32

Table 8: Troisème classification : F1-score des modèles appris sur le jeu d'apprentissage et sur les tweets scientifique du jeu de données d'origine avec la méthode OvR

Comme le montrent les résultats, les performances sont moins bonnes pour le dataset de base. Cette différence peut s'expliquer par l'écart des coefficients de corrélation entre les trois classes sur le jeu d'origine et sur le dataset augmenté. Le dataset augmenté étant plus homogène, les performances des classifieurs sont donc meilleures.

3.3.4 stratégie Classifieur Chaîné multi-classe

Le *Classifier Chain* consiste à entraîner des classifieurs binaires en chaîne, chaque prédiction utilisant les prédictions précédentes comme caractéristiques supplémentaires. Cela permet de modéliser les dépendances entre labels en classification multi-label.

Label	**F1-score**	**Support**
`scientific_claim`	0.53	229
`scientific_reference`	0.52	213
`scientific_context`	0.71	304
Hamming Loss	0.181	
Moyenne pondérée	0.60	(Rappel : 0.50)

Table 9: Résultats de classification multi-label avec ClassifierChain (RandomForest)

Les performances du modèle `ClassifierChain` basé sur `XGBoost` montrent une capacité modérée à identifier les différentes étiquettes scientifiques présentes dans les tweets. Le label `scientific_context` est celui le mieux prédit, probablement en raison de son support plus important. À l'inverse, les labels `scientific_claim` et `scientific_reference` présentent des F1-scores plus faibles.

La *Hamming Loss* globale de 0,18 indique qu'environ 18 % des labels sont incorrectement prédits. Enfin, la *sample average* F1-score (0,60) montre que la prédiction simultanée de tous les labels reste difficile.

3.3.5 stratégie LSTM multi-label

Nous voulions aussi voir ce que pouvait donner un LSTM adapté pour le multi-label.
Le texte a été vectorisé via un `Tokenizer` Keras, suivi d'un `padding` afin d'homogénéiser la longueur des séquences d'entrée. Le jeu de données a été divisé avec train-test-split. Le modèle a ensuite été entraîné sur 15 époques (*epochs*), avec une réévaluation des métriques à chaque itération.

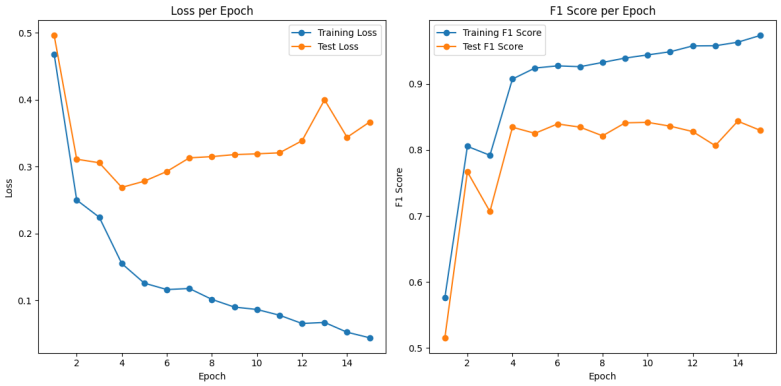


Figure 1: Évolution de la perte et du F1-score au cours des 15 époques d'entraînement pour la classification multi-label avec LSTM.

- **À gauche**, la courbe de la **perte (loss)** montre une décroissance régulière sur le jeu d'entraînement, ce qui est attendu. La perte sur le jeu de validation reste relativement stable après une amélioration initiale, ce qui indique que le modèle généralise correctement sans trop de surapprentissage.
- **À droite**, le **F1-score** augmente fortement pendant les premières époques, pour ensuite se stabiliser. Le F1-score sur l'entraînement continue d'augmenter, mais celui de la validation reste stable autour de 0.83, ce qui suggère une saturation des performances sans overfitting fort.

On a donc de bonnes capacités de généralisation sur la tâche de classification multi-label. Les performances stables entre l'entraînement et la validation indiquent un bon équilibre.

4 Analyses et Conclusions

Après avoir effectué trois classifications différentes et réalisé diverses analyses sur les jeux de données, nous pouvons en tirer les conclusions suivantes:

- La forte ambiguïté présente entre les différentes classes rend difficile la reconnaissance entre *context* et *reference*. De plus, les différences peuvent résider dans des mots ayant les mêmes racines, mais qui sont supprimés lors de la lemmatisation.
- Ce n'est pas une tâche triviale, et de ce fait, les modèles linéaires sont inappropriés mais l'utilisation d'un LSTM non-entraîné n'est pas adapté à nos jeux de données avec une faible capacité de tweets. Même en jouant sur la taille du vecteur de représentation de sortie et le Dropout ou encore le nombre d'epochs on se retrouve vite en sur-apprentissage à cause de l'apprentissage par coeur des neurones.
- L'augmentation du jeu de données amplifie certains traits qui ne sont pas nécessairement dominants dans le jeu de données de base. En conséquence, lors des prédictions externes, les résultats obtenus ne sont pas toujours satisfaisants, car le modèle devient trop spécifique et manque de représentativité, ce qui pourrait s'apparenter à une *hallucination de GPT*.
- Nous avons principalement utilisé TF-IDF pour la vectorisation, adaptée à la faible diversité lexicale du corpus. Toutefois, l'analyse des fréquences par classe a révélé une forte similarité des termes entre classes, suggérant que des méthodes capturant la sémantique, comme Word2Vec ou SciBERT, pouvaient être pertinentes (nous les avons testé brièvement au début de nous travaux mais sans les approfondir). Si SciBERT a montré de bonnes performances, les résultats obtenus avec Word2Vec ont été plus variables.
- Cela étant dit, sur un jeu de données bien équilibré, la seconde stratégie pourrait surpasser la précédente. En effet, la logique même de la stratégie *One-vs-One*, qui exploite les différences fines entre paires de classes, la rend potentiellement plus performante dans un contexte où les classes sont également représentées.

5 Ouverture

Au terme de ce projet, plusieurs pistes d'amélioration se dégagent. Nos expérimentations ont mis en lumière les limites des approches classiques, notamment face à la complexité du langage scientifique sur des formats courts comme les tweets. Malgré les simplifications du corpus, certaines structures linguistiques et corrélations entre labels restent difficilement capturables.

Une première amélioration porterait sur les prétraitements textuels : mieux préserver certaines expressions ou intégrer des techniques de *named entity recognition* enrichirait le contexte linguistique. L'augmentation des données pourrait aussi être affinée par un filtrage sémantique post-génération (via BERTScore) pour en garantir la cohérence.

Côté modélisation, l'usage de modèles comme `SciBERT`, pré-entraînés sur des textes scientifiques, permettrait de mieux saisir la sémantique des classes grâce au *transfert d'apprentissage*. Enfin, des approches plus avancées, telles que les modèles hiérarchiques ou le *multi-task learning*, pourraient renforcer la précision et la généra-lisation des prédictions.

6 Annexe

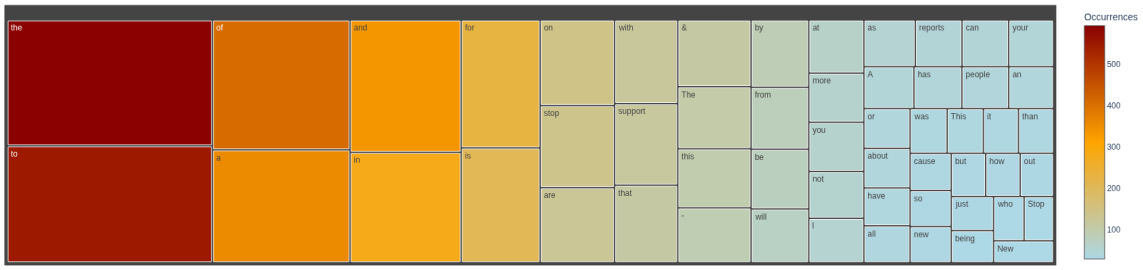
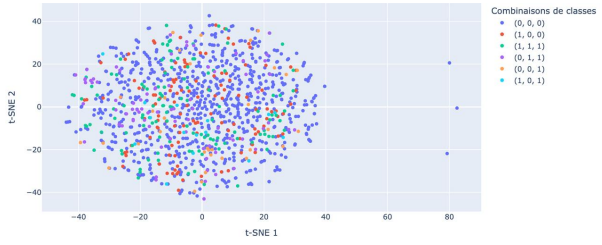
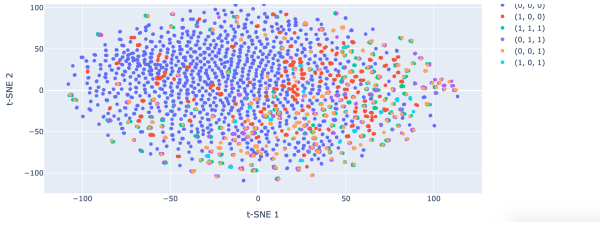


Figure 2: Word Cloud sous forme de Treemap de l'ensemble des tweets de la SciTweets (affichage des mots ayant un nombre d'occurrences supérieur à 30

Figure 3: Word Cloud sous forme de Treemap de l'ensemble des tweets de la SciTweets après prétraitements (affichage des mots ayant un nombre d'occurrences supérieur à 30)



(a) t-SNE du dataset *Scitweet*



(b) t-SNE du dataset *augmenté*

Figure 4: Comparaison des représentations vectorielles t-SNE entre le dataset original et le dataset augmenté.

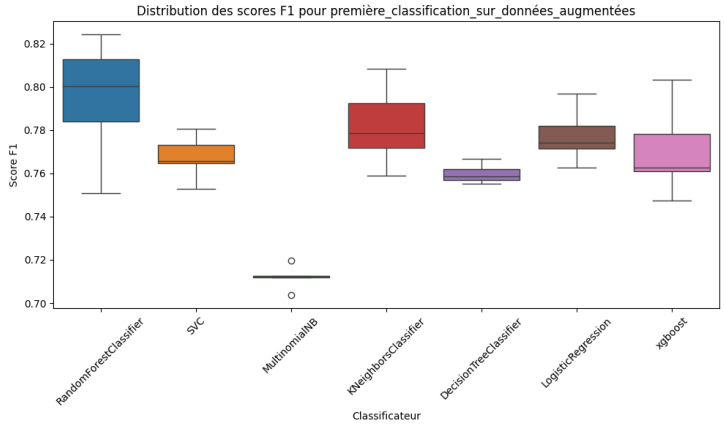


Figure 5: Boîte à moustaches des F1-scores des différents modèles testés par Optuna sur la première classification entraîné et testé sur les données augmentées

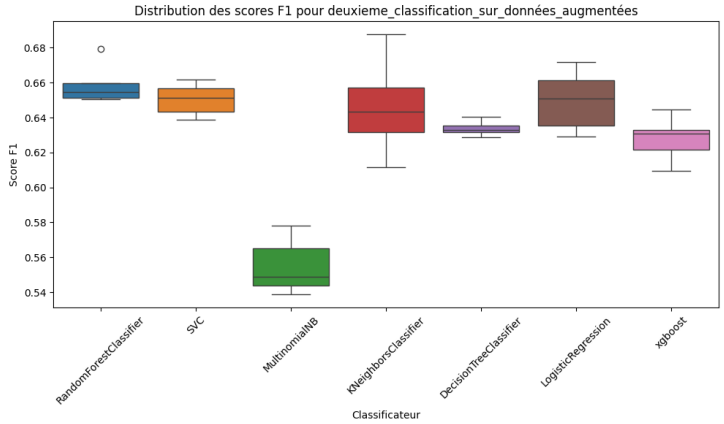


Figure 6: Boîte à moustaches des F1-scores des différents modèles testés par Optuna sur la deuxième classification entraîné et testé sur les données augmentées

7 Bibliographie

References

[1] Todorov, Mihai and Nouvel, Damien and Dufour, Richard, *SciTweets: A Dataset and Annotation Framework for Scientific Discourse Detection in Social Media*

[2] T. Zhang, V. Kishor, K. Weinberger, Y. Artzi and F. Wu, *BERTSscore: Evaluating Text Generation with BERT*

[3] S. Wang, X. Sun, X. Li, R. Ouyang and Y. Wu, *GPT-NER: Named Entity Recognition via Large Language Models*