République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Informatique

Département d'Intelligence Artificielle et Sciences des Données

Projet de Fin de Semestre :

Développement des Classifieurs de Tweets

Module : AARN

Réalisé par

TAOURIRT Hamza

SAADA Samir

Année Universitaire :

2023/2024

Table des matières

Introduction générale

Dans le contexte actuel du monde numérique, les réseaux sociaux jouent un rôle essentiel dans la communication et l'expression des opinions. Les tweets, ces messages courts sur Twitter, reflètent une gamme étendue de sentiments et d'opinions. Notre projet vise à analyser ces sentiments en utilisant les connaissances acquises. Nous développerons un outil pour classifier les tweets en fonction de leur tonalité émotionnelle, que ce soit positive ou négative. Cette démarche implique un processus méthodique, allant de la préparation des données à l'évaluation comparative des différents modèles de classification.

Dans ce contexte, ce projet ambitionne d'exploiter les enseignements théoriques et pratiques acquis au fil du semestre pour aborder un défi essentiel en sciences de données : l'analyse des sentiments à partir de ces tweets.

Dans la première phase de notre projet, nous nous sommes concentrés sur la préparation minutieuse des données, un processus essentiel visant à garantir la qualité et la cohérence des données d'entrée. Cette étape a impliqué le nettoyage des données, la normalisation du texte et la sélection des informations pertinentes, assurant ainsi des bases solides pour la réussite de la classification. Une fois les données préparées, nous avons entrepris la construction d'un vocabulaire exhaustif, regroupant les mots significatifs pour une analyse précise des tweets. Ce vocabulaire a servi de base pour l'extraction des caractéristiques et la représentation des tweets sous forme de vecteurs dans un espace caractéristique, permettant ainsi de capturer efficacement les nuances et les subtilités des messages.

Dans la phase suivante, nous avons exploré divers algorithmes de machine learning pour développer nos classifieurs. Parmi ces méthodes, la régression logistique, les machines à vecteurs de support (SVM) et les arbres de décision, etc. De plus, nous avons également exploré les classifieurs de deep learning, tels que les réseaux de neurones convolutionnels (CNN) et les réseaux de neurones récurrents (RNN), etc. Une fois les modèles entraînés, nous avons effectué une analyse comparative approfondie de leurs performances, mettant en évidence les forces et les faiblesses de chaque approche. Cette analyse a été enrichie par la visualisation des résultats à l'aide de graphiques et de comparaisons détaillées, offrant ainsi un aperçu complet de notre démarche analytique.

1. Préparation des données

1.1 Description du dataset

Le jeu de données Sentiment140 contient 1 600 000 tweets extraits à l'aide de l'API Twitter. Chaque tweet a été annoté selon son sentiment, où 0 représente un sentiment négatif et 4 un sentiment positif, permettant ainsi la détection de sentiment. Le jeu de données se compose de six champs principaux :

	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Third Day to do it. ;D
1599994	4	2193601966	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	AmandaMarie1028	Just woke up. Having no school is the best fee...
1599995	4	2193601969	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	TheWDBoards	TheWDB.com - Very cool to hear old Walt interv...
1599996	4	2193601991	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	bpbabe	Are you ready for your MoJo Makeover? Ask me f...

Figure 1 **:** Segment d'exemple du jeu de données.

- Target : la polarité du tweet (0 pour négatif, 4 pour positif)
- IDs : l'identifiant du tweet
- Date : la date à laquelle le tweet a été publié
- Flag : la requête associée au tweet. Si aucune requête n'est spécifiée, la valeur est NO_QUERY.
- User : le nom d'utilisateur du compte Twitter ayant publié le tweet
- Texte : le contenu textuel du tweet

Le jeu de données est équilibré, avec 800 000 tweets classifiés comme positifs et 800 000 comme négatifs. Pour faciliter l'application de techniques de machine learning, seules les données textuelles ont été conservées.

Hypothèse : Après des tests sur l'ensemble du jeu de données et des sous-ensembles de 700 000 et 100 000 lignes, des performances similaires ont été observées. La variation de la valeur de K a eu un impact sur les performances du modèle. Ainsi, une valeur optimale de K a été choisie pour l'entraînement du classifieur sur un sous-ensemble de 100 000 lignes, équilibré avec 50 000 tweets positifs et 50 000 négatifs.

1.2 Nettoyage et Prétraitement des Tweets

Le nettoyage et le prétraitement des tweets sont essentiels pour garantir la qualité des données utilisées dans l'analyse des sentiments, car ils visent à éliminer les éléments non pertinents ou redondants qui pourraient biaiser les résultats. Ces étapes

incluent la conversion en minuscules pour uniformiser la représentation des mots, la suppression des balises HTML pour éliminer toute information non essentielle, la radicalisation des mots pour réduire leur forme à la racine, et enfin l'élimination de la ponctuation, des non-mots et des mots vides pour simplifier le texte tout en préservant son sens (***Figure 2***). Ce nettoyage initial facilite les étapes ultérieures d'analyse et de classification en fournissant des données textuelles cohérentes et standardisées.

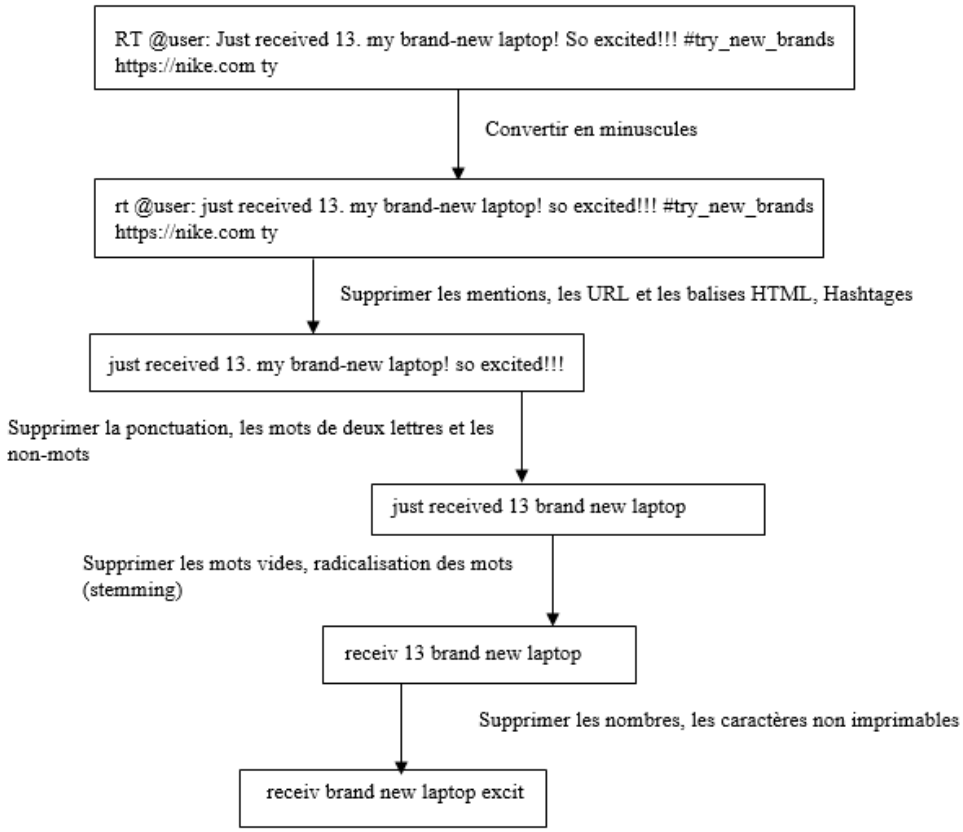

Figure 2 **:** Processus de Prétraitement d'un Tweet.

1.3 Construction de vocabulaire

Après le prétraitement des tweets, une liste de mots représente chaque tweet, filtrée pour inclure uniquement ceux apparaissant au moins K fois. Cette liste est sauvegardée dans "vocab.txt". Ensuite, chaque mot dans les tweets est mappé à son index dans le vocabulaire, permettant une représentation efficace des données pour le modèle de classification.

Pour déterminer la valeur optimale de K, nous avons utilisé l'intégralité (1.6m lignes) du jeu de données pour extraire les caractéristiques les plus significatives. Pour garantir une construction exhaustive du vocabulaire, des tests ont été effectués sur des échantillons de 700 000, 200 000 et 100 000 lignes. Avec K fixé à 1078, nous avons

observé des performances satisfaisantes. En réduisant K en dessous de 1078, les performances ont diminué, tandis qu'une valeur supérieure à 1078 a également entraîné une baisse des performances. Ainsi, une valeur de K égale à 1078 a été retenue pour maximiser l'efficacité du modèle de classification.

```
feel: 19
like: 20
fire: 21
mad: 22
see: 23
need: 24
hug: 25
hey: 26
long: 27
```

Figure 3 : Échantillonnage du Vocabulaire (vocab.txt).

1.4 Extraction de Caractéristiques

Nous avons opté pour une extraction binaire des caractéristiques, permettant de représenter chaque tweet sous forme de vecteur où chaque élément indique la présence (1) ou l'absence (0) d'un mot spécifique dans le tweet. Cette approche transforme les tweets en données numériques exploitables par les algorithmes de classification. Chaque tweet est ainsi représenté par un vecteur de taille fixe, correspondant au nombre de mots uniques dans le vocabulaire, soit 1269 colonnes dans ce cas (***Figure 4***). Cette représentation permet de caractériser chaque tweet par la présence ou l'absence de chaque mot, facilitant ainsi leur analyse et leur classification.

	upset	updat	facebook	text	might	cri	result	school	today	also	...
99995	0	0	0	0	0	0	0	1	0	0	...
99996	0	0	0	0	0	0	0	0	0	0	...

Figure 4 : Échantillonnage du binary featuring.

2. Classification

Une fois les vecteurs caractéristiques obtenus grâce à l'extraction binaire des caractéristiques, nous pouvons procéder à la phase de classification. Nous allons développer des modèles de machine learning tels que la régression logistique, le k-NN (k plus proches voisins), les arbres de décision, etc. De plus, nous envisageons d'explorer des modèles de deep learning, en utilisant les vecteurs caractéristiques binaires comme entrée et en ajustant les classes de manière à ce que la classe 0 devienne 0 et la classe 4 devienne 1, pour les utiliser comme cible lors de l'entraînement des modèles.

2.1 Modèles d'Apprentissage Automatique

2.1.1 La Regression Logistique

L'algorithme de régression logistique modélise la probabilité qu'un tweet appartienne à une classe particulière en utilisant une fonction logistique. En d'autres termes, il évalue la relation entre les variables d'entrée (vecteurs de caractéristiques binaires) et la variable cible (le sentiment du tweet) en calculant les poids des caractéristiques. Ces poids sont ajustés pendant l'entraînement du modèle afin de minimiser l'erreur de prédiction.

Une fois entraîné, le modèle est capable de prédire le sentiment des nouveaux tweets en calculant la probabilité qu'ils appartiennent à la classe positive (4). Ensuite, en utilisant un seuil de décision approprié, les prédictions continues sont converties en étiquettes de classe binaire. Les résultats de la régression logistique sont les suivants :

	Precision	Recall	F1-score	Support
0	0.77	0.71	0.74	10035
4	0.73	0.79	0.76	9965
Accuracy			0.75	20000
Macro avg	0.75	0.75	0.75	20000
Weighted avg	0.75	0.75	0.75	20000

***Tableau 1* :** rapport de classification par Regression logistique.

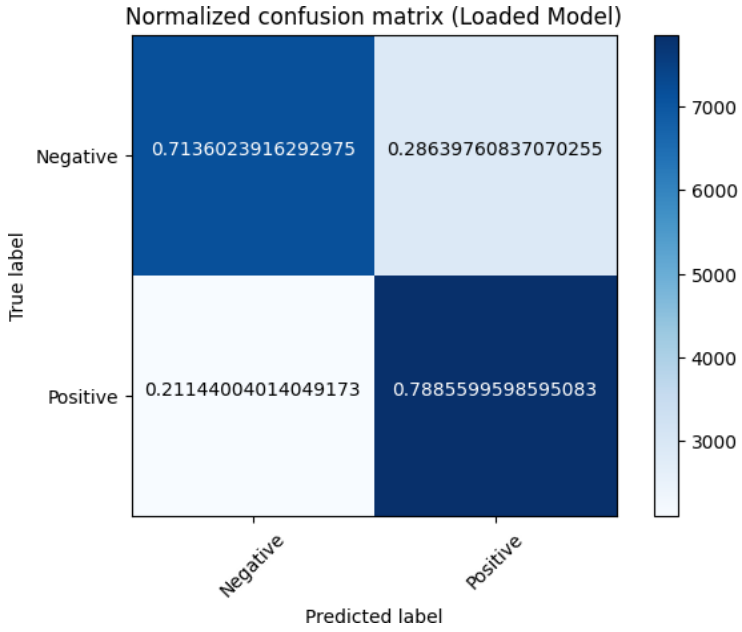

Figure 5 : Matrice de Confusion de la Régression Logistique

Ces résultats montrent une précision globale de 75.10%, avec des performances équilibrées entre les classes 0 et 4 en termes de précision, rappel et score F1. La matrice de confusion révèle également une répartition relativement équilibrée des prédictions correctes et incorrectes pour chaque classe.

2.1.2 La Regression Linéaire

Lorsque nous appliquons la régression linéaire à la prédiction des sentiments des tweets, nous traitons le score de sentiment comme une variable continue, ce qui peut représenter une échelle de sentiment allant de négatif à positif. L'objectif de la régression linéaire est de trouver la meilleure ligne de régression qui minimise l'erreur entre les valeurs prédites et les valeurs réelles du score de sentiment. Cette ligne de régression est déterminée en ajustant les coefficients des caractéristiques à l'aide de méthodes d'optimisation telles que la méthode des moindres carrés.

Pour la régression linéaire, l'utilisation du SGDRegressor a produit des résultats modérés. Avec un Mean Squared Error (**MSE**) de **3.1564** et un Root Mean Squared Error (**RMSE**) de **1.7766**, on observe une dispersion des prédictions par rapport aux valeurs réelles. Le coefficient de détermination (**R-squared ou R2**) de **0.2109** suggère que seulement environ **21.09%** de la variance des données est expliquée par le modèle, indiquant des limitations dans sa capacité à prédire avec précision les sentiments des tweets.

2.1.3 K plus proches voisins (KNN)

L'algorithme des k plus proches voisins (KNN) est utilisé pour la classification binaire des sentiments des tweets. L'idée fondamentale derrière KNN est de classer un point en fonction des classes majoritaires de ses voisins les plus proches dans l'espace de caractéristiques. Pour ce faire, nous utilisons les vecteurs de caractéristiques binaires des tweets comme entrées et les classes réelles (0 ou 4) comme cibles.

L'algorithme KNN nécessite deux paramètres principaux : le nombre de voisins (K=7) à considérer et une mesure de distance pour évaluer la proximité entre les points. Dans ce projet, ces paramètres seront ajustés empiriquement pour obtenir les meilleures performances de classification. Les performances du modèle KNN sont les suivantes :

	Precision	Recall	F1-score	Support
0	0.71	0.60	0.65	10035
4	0.65	0.75	0.70	9965
Accuracy			0.68	20000
Macro avg	0.68	0.68	0.68	20000
Weighted avg	0.68	0.68	0.68	20000

***Tableau 2* :** Rapport de classification par KNN.

Pour la classe 0 (sentiment négatif), le modèle KNN présente un rappel de 0.60, une précision de 0.71 et un f1-score de 0.65. Pour la classe 4 (sentiment positif), le rappel est de 0.75, la précision est de 0.65 et le f1-score est de 0.70. Ces valeurs indiquent une meilleure capacité

du modèle à prédire les sentiments positifs par rapport aux sentiments négatifs, avec des scores globaux modérés pour les deux classes.

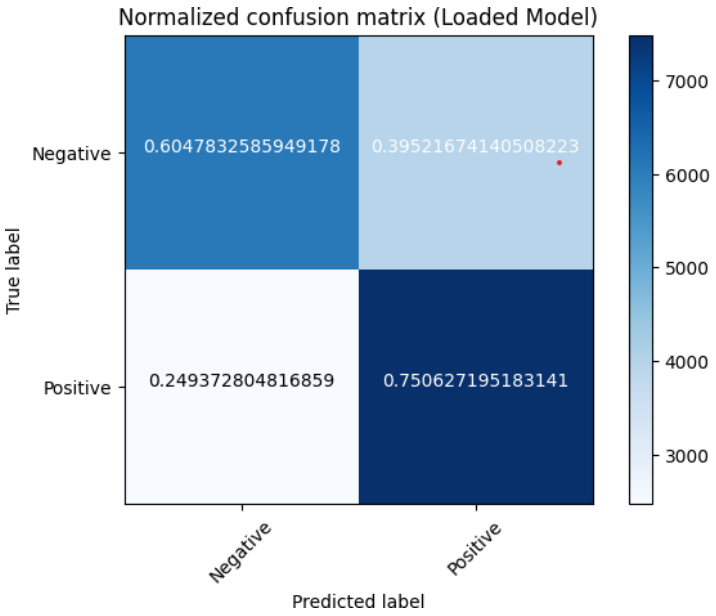

Figure 6 : Matrice de Confusion de KNN.

2.1.4 Arbre de Décision

L'arbre de décision examinera les relations entre les caractéristiques des tweets, identifiant les combinaisons de mots qui sont prédictives des sentiments positifs ou négatifs. Nous examinerons également les performances de l'arbre de décision en termes de précision, de rappel et de f1-score, ainsi que son interprétabilité par rapport à d'autres algorithmes de classification.

	Precision	Recall	F1-score	Support
0	0.71	0.60	0.65	10035
4	0.65	0.75	0.70	9965
Accuracy			0.68	20000
Macro avg	0.68	0.68	0.68	20000
Weighted avg	0.68	0.68	0.68	20000

***Tableau 3* :** Rapport de classification par l'arbre de décision.

Le modèle d'arbre de décision présente une précision globale de 0.6791, ce qui indique une performance modérée dans la classification binaire des sentiments des tweets. Les scores de précision, de rappel et de f1-score pour les classes 0 et 4 sont relativement similaires, indiquant une capacité équilibrée du modèle à prédire les deux classes. Cependant, la matrice de confusion révèle un nombre significatif de faux positifs et de faux négatifs, ce qui suggère que le modèle peut avoir du mal à distinguer les sentiments positifs des négatifs dans certains cas.

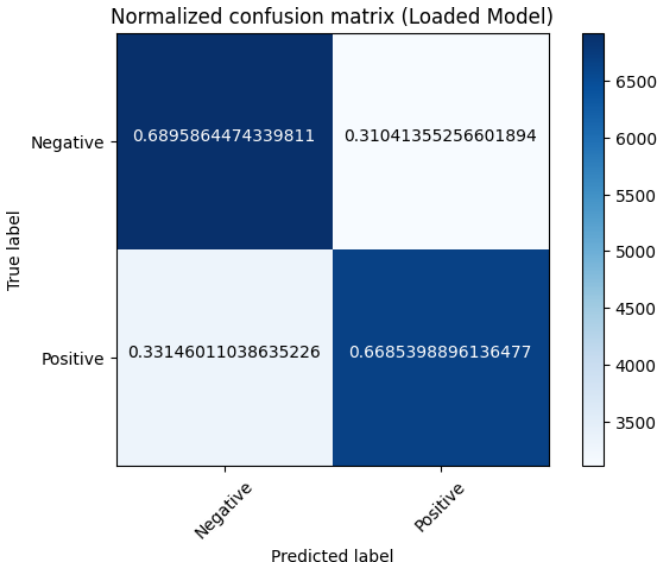

Figure 7 : Matrice de Confusion de l'arbre de décision.

2.1.5 Forêts aléatoires

Les forêts aléatoires traitent les données de caractéristiques binaires de manière similaire aux autres algorithmes d'apprentissage supervisé. Chaque arbre de décision dans la forêt est formé sur un sous-ensemble aléatoire des données d'entraînement et utilise une sélection aléatoire des caractéristiques pour prendre des décisions à chaque nœud (tweet), en fonction de leur présence ou de leur absence. Ce processus est répété pour chaque arbre dans la forêt, et les prédictions finales sont obtenues en agrégeant les prédictions de tous les arbres, souvent par vote majoritaire. Ainsi, les forêts aléatoires peuvent exploiter efficacement les caractéristiques binaires pour classifier les données avec précision et robustesse.

	Precision	Recall	F1-score	Support
0	0.73	0.72	0.73	10035
4	0.72	0.73	0.73	9965
Accuracy			0.73	20000
Macro avg	0.73	0.73	0.73	20000
Weighted avg	0.73	0.73	0.73	20000

***Tableau 4* :** Rapport de classification par les forêts aléatoires.

Les forêts aléatoires ont démontré des performances encourageantes dans la classification binaire des sentiments des tweets, avec une précision globale de 0.7275. Les scores de précision, de rappel et de f1-score pour les classes 0 et 4 sont équilibrés et relativement élevés, tous autour de 0.73, indiquant une capacité équilibrée du modèle à prédire les deux classes avec précision. La matrice de confusion révèle également un nombre de faux positifs et de faux négatifs relativement faible, suggérant une bonne capacité du modèle à distinguer les sentiments positifs des négatifs.

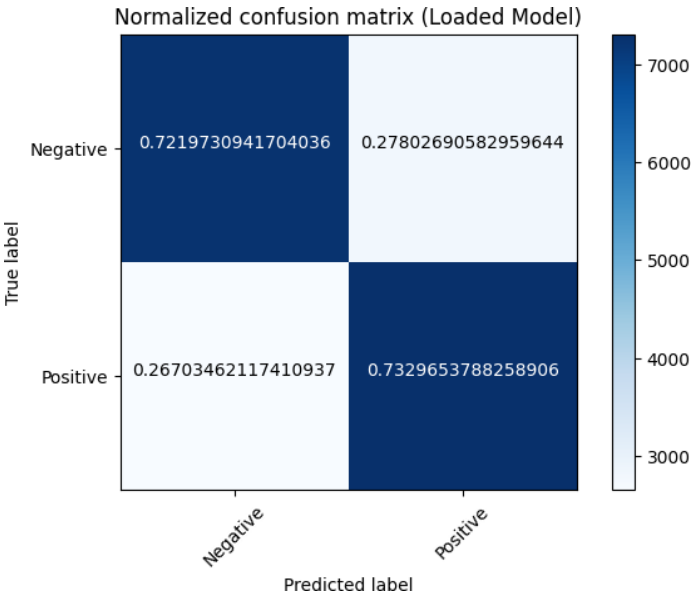

Figure 8 : Matrice de Confusion de l'arbre de décision.

2.1.6 Naive Bayes

L'algorithme Naive Bayes est une méthode probabiliste utilisée pour la classification supervisée. Pour prédire la classe d'un nouveau tweet, Naive Bayes calcule la probabilité conditionnelle de chaque classe donnée les caractéristiques du tweet en utilisant le théorème de Bayes. Dans le cas des vecteurs de caractéristiques binaires des tweets, Naive Bayes suppose que chaque caractéristique est indépendante des autres, simplifiant ainsi le calcul des probabilités conditionnelles. En utilisant ces probabilités, il attribue la classe avec la probabilité la plus élevée comme prédiction pour le nouveau tweet. Ainsi, Naive Bayes est utilisé pour prédire les bonnes classes en se basant sur les probabilités conditionnelles des caractéristiques binaires des tweets.

	Precision	Recall	F1-score	Support
0	0.74	0.60	0.66	10035
4	0.66	0.78	0.72	9965
Accuracy			0.69	20000
Macro avg	0.70	0.69	0.69	20000
Weighted avg	0.70	0.69	0.69	20000

***Tableau 5* :** Rapport de classification par le Naive Bayes.

Le classifieur Naive Bayes a produit des résultats satisfaisants avec une précision globale de 0.6915. Cependant, il montre une tendance à être plus précis dans la prédiction des sentiments positifs (classe 4) par rapport aux sentiments négatifs (classe 0), comme en témoignent les scores de précision, de rappel et de f1-score. Bien que le modèle ait une capacité raisonnable à prédire les deux classes, il peut être nécessaire d'améliorer sa sensibilité à la classe minoritaire (négative) pour une meilleure performance globale.

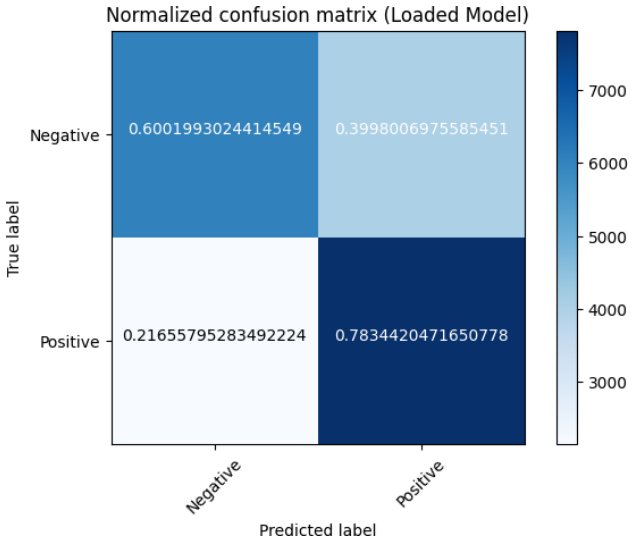

Figure 9 : Matrice de Confusion de Naive Bayes.

2.1.7 Support Vector Machine (SVM)

Le Support Vector Machine (SVM) prédit les classes des tweets en trouvant l'hyperplan qui sépare au mieux les deux classes dans l'espace des caractéristiques binaires des tweets. Lors de l'entraînement, le SVM ajuste cet hyperplan en maximisant la marge, c'est-à-dire la distance entre l'hyperplan et les échantillons d'entraînement les plus proches de chaque classe, appelés vecteurs de support.

En étudiant les relations entre les caractéristiques en entrée, le SVM cherche à trouver un hyperplan qui optimise la séparation entre les deux classes de tweets, en minimisant le risque de classification erronée. Il prend en compte les vecteurs de caractéristiques binaires pour construire cet hyperplan de séparation, assurant ainsi une classification précise des tweets en fonction de leur sentiment.

	Precision	Recall	F1-score	Support
0	0.53	0.66	0.59	10035
4	0.54	0.41	0.47	9965
Accuracy			0.54	20000
Macro avg	0.54	0.53	0.53	20000
Weighted avg	0.54	0.54	0.53	20000

***Tableau 6* :** Rapport de classification par le SVM.

Ce résultat montre que le modèle SVM atteint une précision globale de 53.51%. Pour la classe 0 (sentiment négatif), la précision est de 53% et le rappel (recall) est de 66%, tandis que pour la classe 4 (sentiment positif), la précision est de 54% et le rappel est de 41%. Ces chiffres indiquent que le modèle a du mal à généraliser et à prédire avec précision les deux classes de sentiments. La matrice de confusion révèle une quantité importante de faux positifs et de faux négatifs, montrant que le modèle SVM ne parvient pas à séparer efficacement les deux classes.

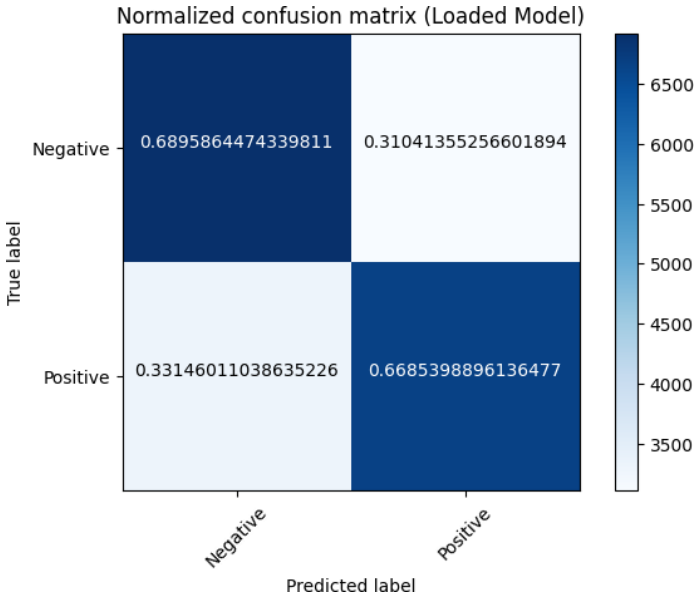

Figure 10 : Matrice de Confusion de SVM.

2.2 Modèles d'Apprentissage Profond

2.2.1 Réseaux de Neurones Artificiels (ANN)

Les Réseaux de Neurones Artificiels (ANN) vont prédire les classes des tweets en apprenant des patterns à partir des vecteurs de caractéristiques binaires. Chaque neurone dans un ANN est connecté à plusieurs autres neurones, formant ainsi des couches de traitement de l'information. Lors de l'entraînement, les poids des connexions entre les neurones sont ajustés itérativement pour minimiser l'erreur de prédiction sur un ensemble d'entraînement. Cette optimisation permet aux ANN de découvrir et de modéliser les relations complexes entre les caractéristiques des tweets et les classes de sentiment associées. En utilisant des algorithmes de rétropropagation du gradient, les ANN ajustent ces poids pour maximiser leur capacité à prédire correctement les classes de sentiment, tout en évitant le surapprentissage aux données d'entraînement. Le tableau 7 illustre les paramètres de Modèle ANN :

Paramètres	Valeur
Activation	Relu, Sigmoïde
Optimizer	Adam
Learning rate	0.001
Loss	Binaire_crossentropy
Epochs	15
Batch size	512
Validation split	0.2
Number of layers	3 de type Dense
Regularization	regularizers.l2(0.01)

***Tableau 7* :** Les différents paramètres de modèle ANN.

Accuracy	Loss	Recall	F1-score	Precision
0.7384	0.5913	0.7722	0.7463	0.7221

***Tableau 8* :** Les différents métriques de modèle ANN.

Le modèle ANN a atteint une précision de test de 73.84% avec une perte de 0.5913. Sur l'ensemble de test, la précision, le rappel et le score F1 sont respectivement de 72.21%, 77.22%, et 74.63%. Cela signifie que le modèle est capable de prédire avec précision les classes de sentiment des tweets.

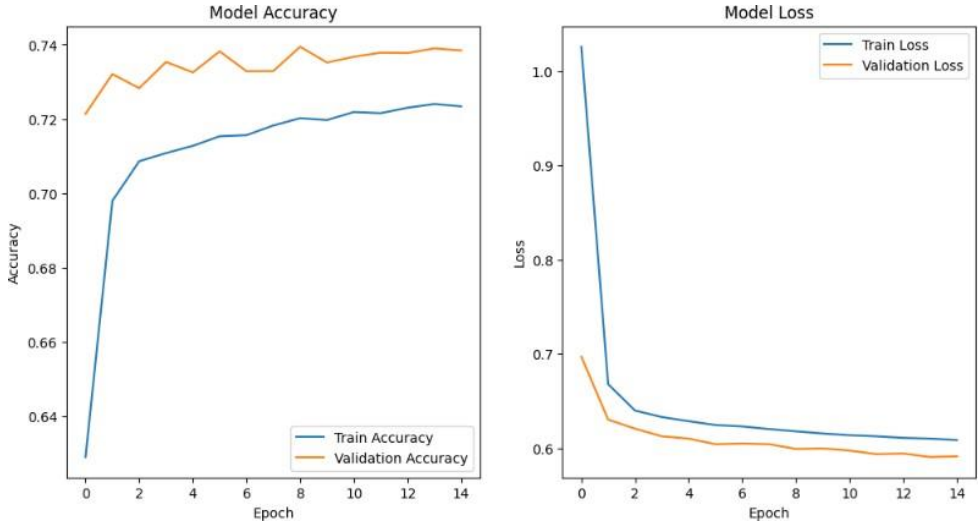

Figure 11 : L'évolution de l'accuracy et loss pour ANN.

2.2.2 Réseaux de Neurones récurrents (RNN)

Les Réseaux de Neurones Récurrents (RNN) prédisent les classes des tweets en traitant séquentiellement les vecteurs de caractéristiques binaires des mots dans chaque tweet. Ils utilisent une architecture de réseau neuronal récurrente pour capturer les dépendances séquentielles entre les mots dans le tweet. Cependant, étant donné que les caractéristiques binaires ne présentent pas de dépendance temporelle entre elles, nous utilisons plutôt des couches denses (Dense Layers) comme couches d'entrée pour accélérer l'exécution du modèle. En effet, les RNN sont positionnés en couches cachées étudient ainsi les relations entre les caractéristiques en analysant la séquence de mots dans chaque tweet et en apprenant à partir de ces informations pour prédire la classe de sentiment correspondante. Le tableau suivant illustre les paramètres de RNN :

Paramètres	Valeur
Activation	Relu, Sigmoïde
Optimizer	Adam
Learning rate	0.001
Loss	Binaire_crossentropy
Epochs	15
Batch size	512
Validation split	0.2
Number of layers	1 Dense en entrée, 2 SimpleRNN, 1 Dense en sortie
Regularization	regularizers.l2(0.01)

***Tableau 9* :** Les différents paramètres de modèle RNN.

Accuracy	Loss	Recall	F1-score	Precision
0.7441	0.5633	0.7854	0.7536	0.7243

***Tableau 10* :** Les différents métriques de modèle RNN.

La précision de 74,41 % indique le pourcentage de prédictions correctes parmi toutes les classifications effectuées. Le score F1 de 75,36 % est une mesure harmonique de la précision et du rappel, donnant une indication globale de l'exactitude des prédictions du modèle. Un rappel de 78,54 % indique la capacité du modèle à identifier correctement les vrais positifs parmi tous les exemples positifs dans les données. Enfin, la précision de 72,43 % indique la proportion de vrais positifs parmi toutes les prédictions positives faites par le modèle. Ces performances confirment leur capacité à analyser efficacement les données binaires et à prédire avec précision si un tweet appartient à la classe 0 ou 1.

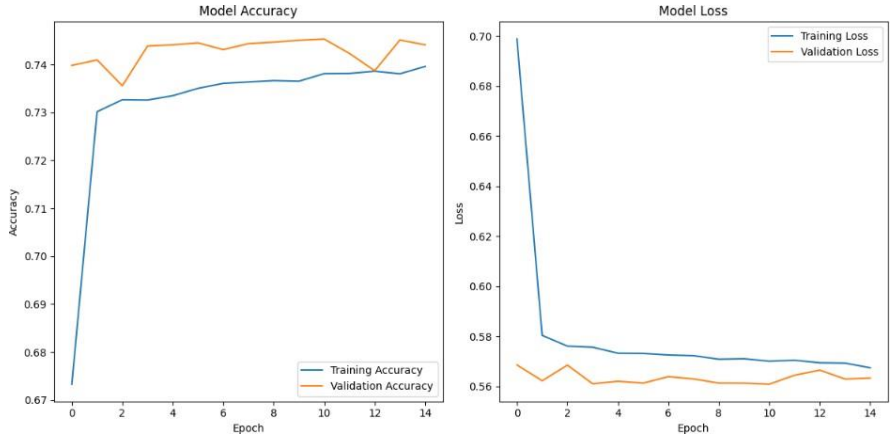

Figure 12 : L'évolution de l'accuracy et loss pour RNN.

2.2.3 Réseaux de Neurones (LSTM)

Les réseaux de neurones LSTM (Long Short-Term Memory) sont une architecture de réseau de neurones récurrents (RNN) particulièrement efficace pour traiter les données séquentielles. Ils sont capables de capturer les dépendances à long terme dans les séquences de données, ce qui en fait un choix populaire pour l'analyse de texte et d'autres tâches impliquant des séquences. Dans notre contexte, les LSTM seront utilisés pour prédire les classes des tweets en se basant sur les vecteurs de caractéristiques binaires. Ces réseaux examinent les relations entre les caractéristiques binaires pour détecter les motifs significatifs dans les données.

Accuracy	Loss	Recall	F1-score	Precision
0.7429	0.5648	0.8021	0.7566	0.7160

***Tableau 11* :** Les différents métriques de modèle lSTM.

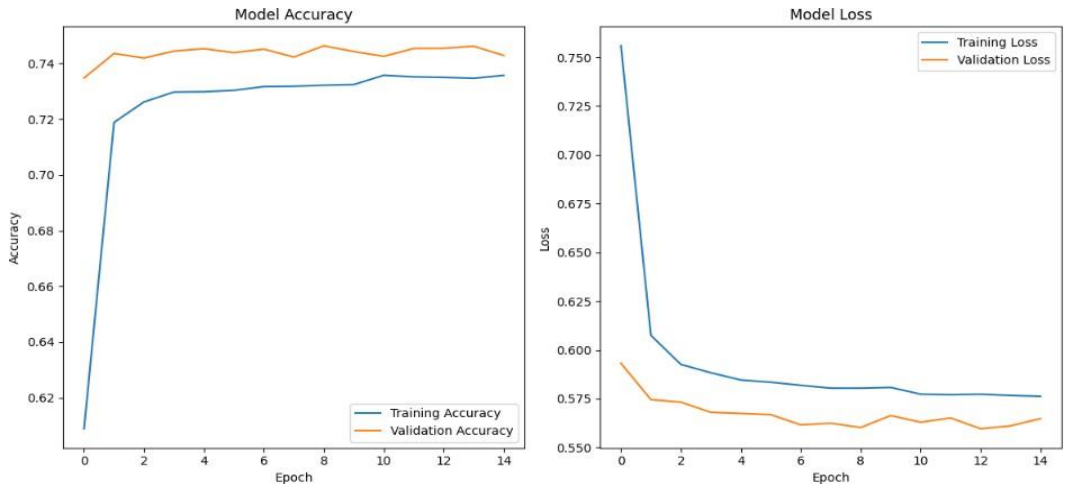

Figure 13 : L'évolution de l'accuracy et loss pour LSTM.

Les performances du modèle LSTM sont solides, avec une précision de 74,29% et un score F1 de 75,66%. Cela indique une capacité satisfaisante à prédire avec précision les classes des tweets. De plus, un rappel élevé de 80,21% suggère que le modèle est capable de capturer une grande proportion des exemples positifs dans les données. Cependant, la précision de 71,60% montre qu'il existe encore une marge d'amélioration pour réduire les faux positifs, par à rapport au RNN.

2.2.4 Les réseaux de neurones convolutifs CNN (CONV1D)

Conv1D, une variante de CNN, est particulièrement efficace pour l'analyse de séquences unidimensionnelles telles que les données textuelles. En appliquant des filtres de convolution unidimensionnels sur les vecteurs de caractéristiques binaires

représentant les tweets, Conv1D peut capturer les motifs locaux et globaux importants dans le tweet, ce qui en fait un choix prometteur pour la classification de tweets.

Accuracy	Loss	Recall	F1-score	Precision
0.7228	0.5965	0.7385	0.7264	0.7147

***Tableau 12* :** Les différents métriques de modèle CNN (CONV1D).

Le modèle CNN (conv1d) présente une précision de test de 72,28%, avec une précision de 71,47%, un rappel de 73,85% et un score F1 de 72,64%. Ces résultats indiquent une capacité modérée à identifier avec précision les classes des tweets.

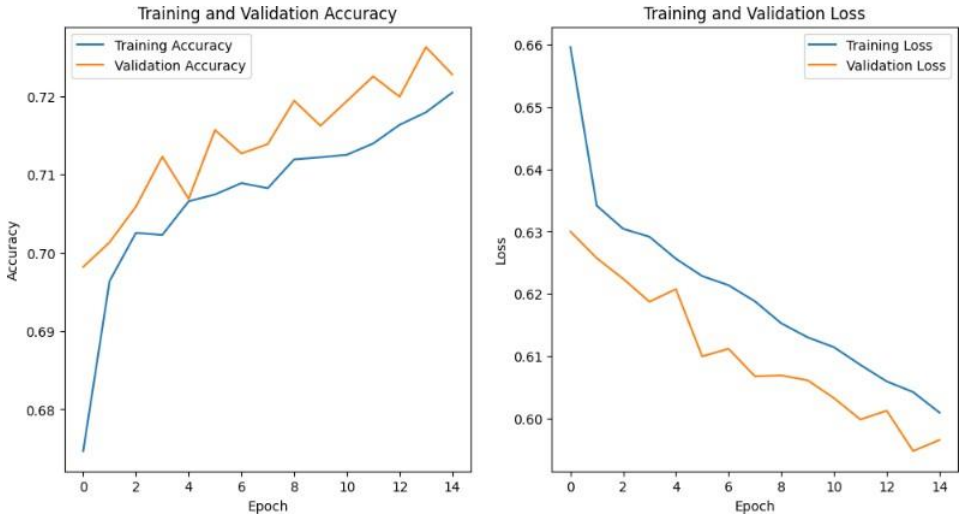

Figure 14 : L'évolution de l'accuracy et loss pour CNN.

2.2.5 Réseaux de Neurones (BI-LSTM)

Les BI-LSTM peuvent prédire les classes des tweets 0 ou 1 en utilisant leurs couches récurrentes bidirectionnelles pour capturer les relations entre les caractéristiques binaires. Ces réseaux exploitent des mécanismes de mémoire à long terme pour encoder les informations des caractéristiques binaires et examiner séquentiellement les relations entre elles. En utilisant des couches LSTM bidirectionnelles, les BI-LSTM peuvent prendre en compte les informations passées et futures lors de la prédiction, ce qui leur permet de mieux comprendre le contexte global des données et d'effectuer des classifications précises.

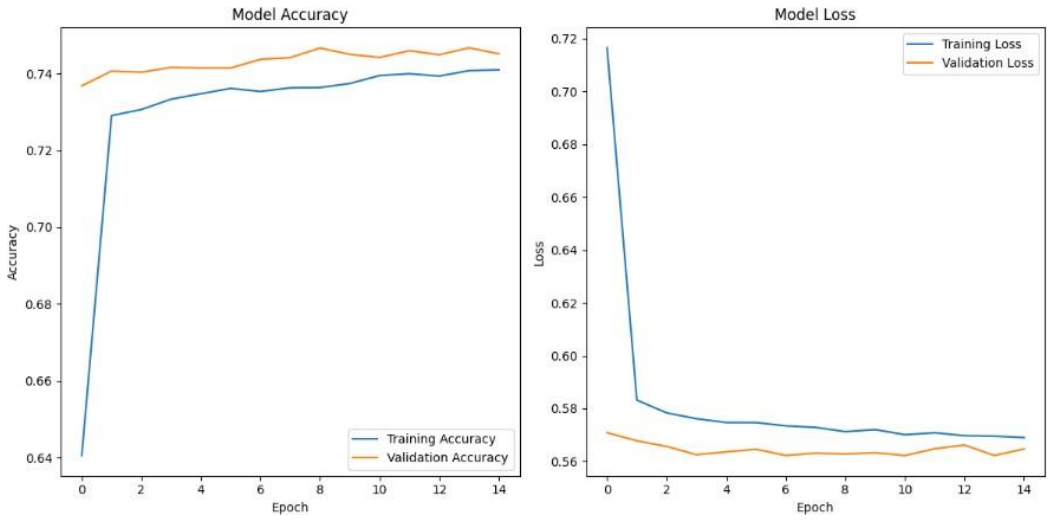

Figure 15 : L'évolution de l'accuracy et loss pour BI-LSTM.

Accuracy	Loss	Recall	F1-score	Precision
0.7452	0.5647	0.7909	0.7556	0.7234

***Tableau 13* :** Les différents métriques de modèle BI-LSTM.

Les résultats des BI-LSTM sont assez prometteurs, avec une précision de test de 74,52 % et un score F1 de 75,56 %. Cela suggère une capacité significative à prédire avec précision les classes des tweets. En outre, le rappel élevé de 79,09 % indique que les BI-LSTM sont efficaces pour identifier correctement les instances positives.

2.2.6 Gated Recurrent Unit (GRU)

Les réseaux de neurones GRU (Gated Recurrent Unit) sont utilisés pour prédire les classes des tweets en utilisant les vecteurs de caractéristiques binaires en entrée. Ils examinent les relations entre ces caractéristiques en utilisant des mécanismes internes de rétroaction. Les GRU sont conçus pour capturer les dépendances temporelles dans les données séquentielles, mais dans ce cas où les caractéristiques binaires ne présentent pas de dépendance temporelle, les GRU peuvent être utilisés avec une architecture adaptée pour analyser les relations entre les différentes caractéristiques.

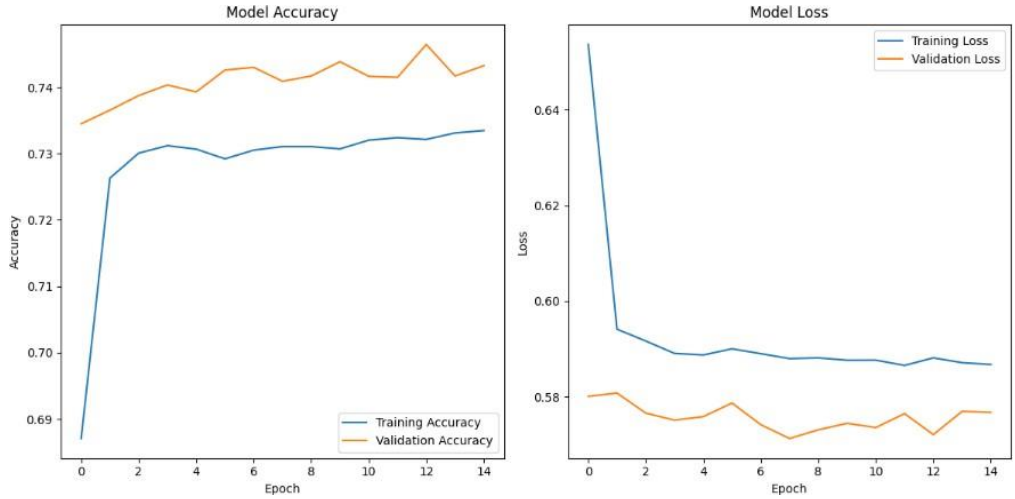

Figure 16 : L'évolution de l'accuracy et loss pour GRU.

Accuracy	Loss	Recall	F1-score	Precision
0.7409	0.5713	0.7455	0.7414	0.7374

***Tableau 14* :** Les différents métriques de modèle GRU.

Les GRU affichent une précision de test de 74,09 % avec une perte de 0,5713. Le rappel est de 74,55 %, et le score F1 est de 74,14 %. La précision atteint 73,74 %, soulignant ainsi une performance globale respectable dans la prédiction des classes des tweets.

2.2.7 CNN (CONV1D) Combiné avec les BI-LSTM

Le modèle combiné Conv1D-BI-LSTM commence par les couches de convolution 1D, qui analysent les motifs locaux dans les caractéristiques binaires des tweets. Ces motifs sont capturés à l'aide de filtres de convolution qui se déplacent le long des séquences de caractéristiques binaires, extrayant des informations pertinentes à partir de fenêtres de données. Ensuite, les sorties de ces couches de convolution sont transmises aux couches BI-LSTM, qui explorent les dépendances à long terme entre les caractéristiques. Les réseaux BI-LSTM sont capables de capturer les relations complexes entre les mots dans les tweets. Enfin, les sorties des couches BI-LSTM sont agrégées et transmises à une couche dense finale, qui effectue la classification binaire des tweets en prédissant les classes 0 ou 1.

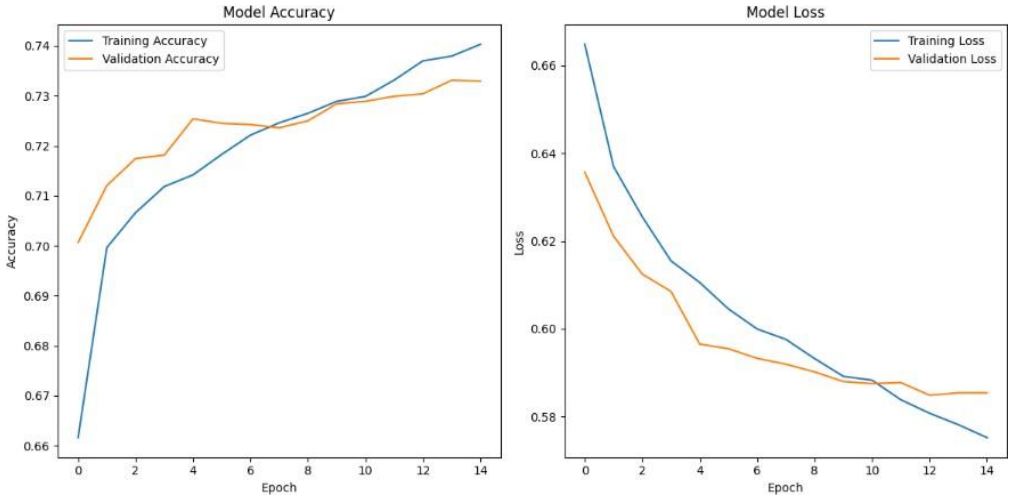

Figure 17 : L'évolution de l'accuracy et loss pour Conv1D-BI-LSTM.

Accuracy	Loss	Recall	F1-score	Precision
0.7329	0.5713	0.7396	0.7340	0.7285

***Tableau 15* :** Les différents métriques de modèle Conv1D-BI-LSTM.

Le modèle combiné Conv1D-BI-LSTM a démontré des performances encourageantes dans la prédiction des classes des tweets. Avec une précision de test de 73,29%, ce modèle présente une capacité satisfaisante à classifier les tweets dans les bonnes catégories. La précision, le rappel et le score F1 du modèle sont également équilibrés, ce qui suggère une bonne capacité à identifier à la fois les tweets de la classe 0 et de la classe 1. Cela indique que la combinaison de Conv1D et de BI-LSTM a permis d'exploiter efficacement les caractéristiques des tweets binaires pour une classification précise.

3. Comparaison des Modèles

Pour la comparaison entre les différents modèles de ML et DL, le tableau ci-dessous présente les performances obtenues par chaque modèle :

Classifieur \ classe	Precision	Recall	F1-score	Accuracy
Régression logistique \ 0	0.77	0.71	0.74	0.7510
Régression logistique \ 4	0.73	0.79	0.76	
KNN \ 0	0.71	0.60	0.65	0.6774
KNN \ 4	0.65	0.75	0.70	

L'arbre de décision \ 0	0.71	0.60	0.65	0.6791
L'arbre de décision \ 4	0.65	0.75	0.70	
Les forêts aléatoires \ 0	0.73	0.72	0.73	0.7275
Les forêts aléatoires \ 4	0.72	0.73	0.73	
Naive Bayes \ 0	0.74	0.60	0.66	0.6915
Naive Bayes \ 4	0.66	0.78	0.72	
SVM \ 0	0.53	0.66	0.59	0.5351
SVM \ 4	0.54	0.41	0.47	
ANN \ 0	0.76	0.70	0.73	0.7384
ANN \ 4	0.72	0.77	0.75	
RNN \ 0	0.77	0.70	0.73	0.7441
RNN \ 4	0.72	0.79	0.75	
LSTM \ 0	0.78	0.68	0.73	0.7429
LSTM \ 4	0.72	0.80	0.76	
CNN \ 0	0.73	0.71	0.72	0.7228
CNN \ 4	0.71	0.74	0.73	
BI- LSTM \ 0	0.77	0.70	0.73	0.7452
BI- LSTM \ 4	0.72	0.79	0.76	
GRU \ 0	0.74	0.74	0.74	0.7409
GRU \ 4	0.74	0.75	0.74	
Conv1D-BI-LSTM \ 0	0.74	0.73	0.73	0.7329
Conv1D-BI-LSTM \ 4	0.73	0.74	0.73	

***Tableau 16* :** Comparaison des performances des différents modèles de classification.

La régression logistique a démontré des performances solides, avec une exactitude globale de 75,10%. Pour la classe 0, elle a obtenu une précision de 77%, un rappel de 71% et un score F1 de 74%. Quant à la classe 4, les résultats étaient légèrement meilleurs avec une précision de 73%, un rappel de 79% et un score F1 de 76%. La régression logistique semble donc bien équilibrée pour classer les deux classes.

Les performances des K-plus proches voisins (KNN) étaient nettement inférieures avec une exactitude globale de 67,74%. Pour la classe 0, le modèle a obtenu une précision de

71% et un rappel de 60%, ce qui se traduit par un faible score F1 de 65%. La classe 4 a été mieux classée avec une précision de 65%, un rappel de 75% et un score F1 de 70%. Le faible rappel de 60% pour la classe 0 indique que le modèle KNN a eu du mal à détecter correctement de nombreuses instances de cette classe.

Arbre de Décision, tout comme le modèle KNN, l'arbre de décision a rencontré des difficultés, atteignant une exactitude globale de seulement 67,91%. Ses performances pour la classe 0 étaient identiques au KNN avec une précision de 71%, un rappel de 60% et un score F1 de 65%. Pour la classe 4, l'arbre a obtenu une précision de 65%, un rappel de 75% et un score F1 de 70%.

Les forêts aléatoires se sont avérées être un choix beaucoup plus performant avec une exactitude globale de 72,75%. Pour la classe 0, elles ont atteint une précision de 73%, un rappel de 72% et un score F1 de 73%. La classe 4 a été classée de manière similaire avec 72% de précision, 73% de rappel et un score F1 de 73%. Les forêts aléatoires semblent donc bien équilibrées pour les deux classes.

Le classifieur Naive Bayes a obtenu une exactitude globale de 69,15%. Bien qu'ayant une précision décente de 74% pour la classe 0, son rappel n'était que de 60%, résultant en un faible score F1 de 66%. Pour la classe 4, les résultats étaient meilleurs avec 66% de précision, 78% de rappel et un score F1 de 72%.

Le modèle SVM a été le moins performant avec une exactitude globale de seulement 53,51%, bien que sa précision pour la classe 0 (53%) et son rappel pour la classe 4 (41%) soient parmi les plus faibles de tous les modèles.

Réseaux de Neurones (ANN, RNN, LSTM, GRU, CNN, BI-LSTM) : Les différentes architectures de réseaux de neurones ont globalement démontré de bonnes performances. Les réseaux convolutionnels 1D (CNN) ont obtenu 73% de précision et 71% de rappel pour la classe 0 (F1 72%), et 71% de précision avec 74% de rappel pour la classe 4 (F1 73%), pour une exactitude globale de 72,28%. Les réseaux récurrents simples (RNN) ont légèrement dépassé les CNN avec 77% de précision et 70% de rappel pour la classe 0 (F1 73%), et 72% de précision avec 79% de rappel pour la classe 4 (F1 75%). Leur exactitude globale était de 74,41%. Les réseaux LSTM ont fait encore mieux avec 78% de précision pour la classe 0 (rappel 68%, F1 73%) et 72% de précision avec 80% de rappel pour la classe 4 (F1 76%). Leur exactitude globale était de 74,29%. Les GRU ont été légèrement inférieurs aux LSTM avec une exactitude de 74,09%. Enfin, les BI-LSTM ont été les plus performants parmi les réseaux de neurones avec 77% de précision et 70% de rappel pour la classe 0 (F1 73%), et 72% de précision avec 79% de rappel pour la classe 4 (F1 76%). Leur exactitude globale était de 74,52%.

Ces résultats montrent que les modèles de réseaux de neurones, en particulier les RNN et BI-LSTM, ainsi que les méthodes traditionnelles telles que la régression logistique ont

tendance à surpasser les autres approches en termes de performances de classification pour ces données spécifiques. Ce qui était confirmé par la courbe ROC suivante :

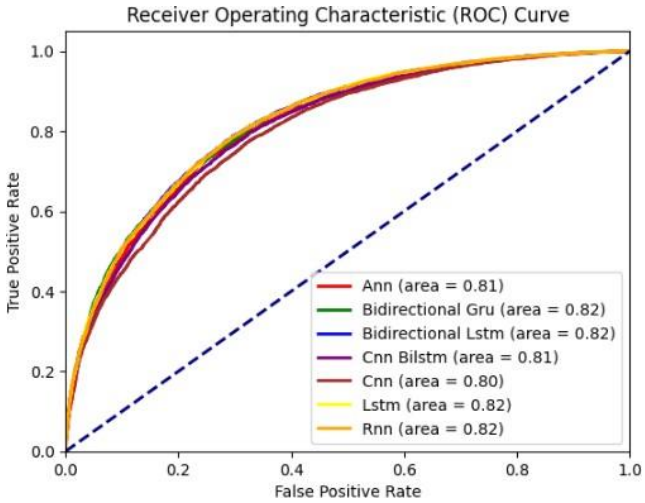

Figure 18 : Courbe ROC des Modèles de Deep Learning.

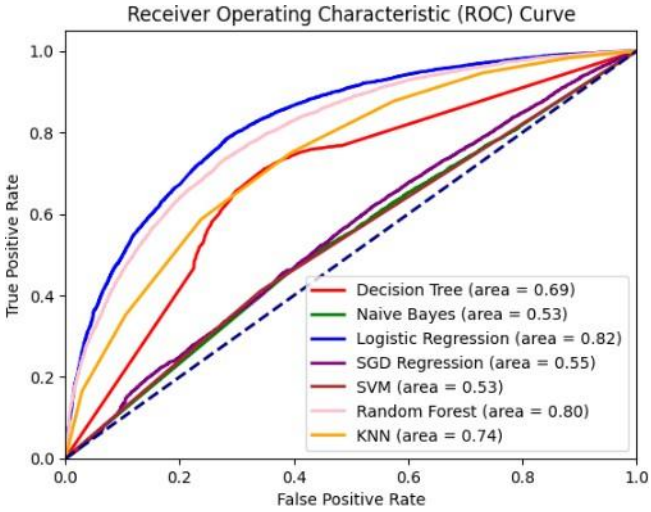

Figure 19 : Courbe ROC des Modèles de Machine Learning.

4. Tests des Modèles

Une fonction predict_tweet_sentiment(tweet) a été développée pour prédire le sentiment d'un tweet. Cette fonction commence par prétraiter le tweet, puis le convertit en un vecteur de caractéristiques binaires. Ensuite, elle charge différents modèles de machine learning préalablement entraînés, tels que l'arbre de décision, la régression logistique, le Naive Bayes, le SVM, etc. Pour chaque modèle, la fonction effectue une prédiction du sentiment du tweet et affiche le résultat. Les figurent suivantes illustrent le résultat obtenu :

- **Pour un tweet complique et considéré comme positif :** « I had the most wonderful day today! The weather was perfect, and I spent the afternoon with my friends, laughing and enjoying each other's company. Later, we went for a delicious dinner at our favorite restaurant, and the food was absolutely amazing. I feel so grateful for moments like these, filled with joy and happiness. It's days like this that remind me how beautiful life can be. #blessed #grateful »

```
# Example usage:
tweet_to_predict = "I had the most wonderful day today! The weather wa
ML_predict_tweet_sentiment(tweet_to_predict)

The tweet is classified as negative by Decision Tree.
The tweet is classified as positive by Logistic Regression.
The tweet is classified as negative by Naive Bayes.
The tweet is classified as positive by SGD Regression.
The tweet is classified as negative by SVM.
The tweet is classified as positive by Random Forest Classifier.
The tweet is classified as positive by KNN.
```

Figure 20 : Test sur un Tweet Positive compliqué Par Les modèles de ML.

On remarque que seulement la régression logistique, les forêts aléatoires, KNN et SGD ont prédit la classe correctement, ce qui était évident d'après leurs résultats.

```
# Example usage:
tweet_to_predict = "I had the most wonderful day today! The
predict_tweet_sentiment(tweet_to_predict)

1/1 [==============================] - 0s 107ms/step
The tweet is classified as positive by Ann.
1/1 [==============================] - 2s 2s/step
The tweet is classified as positive by Bidirectional Gru.
1/1 [==============================] - 2s 2s/step
The tweet is classified as positive by Bidirectional Lstm.
1/1 [==============================] - 2s 2s/step
The tweet is classified as positive by Cnn Bilstm.
1/1 [==============================] - 0s 80ms/step
The tweet is classified as positive by Cnn.
1/1 [==============================] - 0s 480ms/step
The tweet is classified as positive by Lstm.
1/1 [==============================] - 0s 303ms/step
The tweet is classified as positive by Rnn.
```

Figure 20 : Test sur un Tweet Positive compliqué Par Les modèles de DL.

On remarque que les modèles de deep learning ont bien prédit la classe.

- **Pour un tweet complique et considéré comme négative :** « Feeling a bit under the weather today. Nothing seems to be going right. #MondayBlues »

```
# Example usage:
tweet_to_predict = "Feeling a bit under the weather today. Nothing se
ML_predict_tweet_sentiment(tweet_to_predict)

The tweet is classified as negative by Decision Tree.
The tweet is classified as negative by Logistic Regression.
The tweet is classified as positive by Naive Bayes.
The tweet is classified as positive by SGD Regression.
The tweet is classified as negative by SVM.
The tweet is classified as negative by Random Forest Classifier.
The tweet is classified as negative by KNN.
```

Figure 22 : Test sur un Tweet Négative compliqué Par Les modèles de ML.

On remarque que seulement la régression logistique, l'arbre de décision, les forêts aléatoires, KNN et SGD ont prédit la classe correctement, ce qui était évident d'après leurs résultats, ainsi que d'autres modèles tel que SVM et l'arbre de décision.

```
# Example usage:
tweet_to_predict = "Feeling a bit under the weather today. Nothing
predict_tweet_sentiment(tweet_to_predict)

1/1 [==============================] - 0s 66ms/step
The tweet is classified as negative by Ann.
1/1 [==============================] - 2s 2s/step
The tweet is classified as negative by Bidirectional Gru.
1/1 [==============================] - 2s 2s/step
The tweet is classified as negative by Bidirectional Lstm.
1/1 [==============================] - 2s 2s/step
The tweet is classified as negative by Cnn Bilstm.
1/1 [==============================] - 0s 80ms/step
The tweet is classified as negative by Cnn.
1/1 [==============================] - 0s 478ms/step
The tweet is classified as negative by Lstm.
1/1 [==============================] - 0s 282ms/step
The tweet is classified as negative by Rnn.
```

Figure 23 : Test sur un Tweet Négative compliqué Par Les modèles de DL.

Les modèles de DL ont des performances prometteuses car ils ont tous prédit la classe correctement.

- **Pour un tweet simple et considéré comme négative :** « I hate apple »

```
# Example usage:
tweet_to_predict = "I hate apple"
ML_predict_tweet_sentiment(tweet_to_predict)

The tweet is classified as negative by Decision Tree.
The tweet is classified as negative by Logistic Regression.
The tweet is classified as positive by Naive Bayes.
The tweet is classified as positive by SGD Regression.
The tweet is classified as positive by SVM.
The tweet is classified as negative by Random Forest Classifier.
The tweet is classified as negative by KNN.
```

Figure 24 : Test sur un Tweet Négative Simple par Les modèles de ML.

On remarque que le SVM, la régression linéaire (SGD Regression) et le Naive Bayes ont donné de faux résultats, comme en témoigne leur mauvaise performance.

```
# Example usage:
tweet_to_predict = "I hate apple"
predict_tweet_sentiment(tweet_to_predict)

1/1 [==============================] - 0s 105ms/step
The tweet is classified as negative by Ann.
1/1 [==============================] - 2s 2s/step
The tweet is classified as negative by Bidirectional Gru.
1/1 [==============================] - 2s 2s/step
The tweet is classified as negative by Bidirectional Lstm.
1/1 [==============================] - 2s 2s/step
The tweet is classified as negative by Cnn Bilstm.
1/1 [==============================] - 0s 124ms/step
The tweet is classified as negative by Cnn.
1/1 [==============================] - 1s 753ms/step
The tweet is classified as negative by Lstm.
1/1 [==============================] - 0s 413ms/step
The tweet is classified as negative by Rnn.
```

Figure 25 : Test sur un Tweet Négative Simple par Les modèles de DL.

Les modèles de deep learning ont toujours prédit les classes correctement.

- **Pour un tweet simple et considéré comme positive :** « I like apple »

```
# Example usage:
tweet_to_predict = "I like apple"
predict_tweet_sentiment(tweet_to_predict)
```

```
1/1 [==============================] - 0s 88ms/step
The tweet is classified as positive by Ann.
1/1 [==============================] - 3s 3s/step
The tweet is classified as positive by Bidirectional Gru.
1/1 [==============================] - 2s 2s/step
The tweet is classified as positive by Bidirectional Lstm.
1/1 [==============================] - 2s 2s/step
The tweet is classified as positive by Cnn Bilstm.
1/1 [==============================] - 0s 171ms/step
The tweet is classified as positive by Cnn.
1/1 [==============================] - 1s 820ms/step
The tweet is classified as positive by Lstm.
1/1 [==============================] - 1s 560ms/step
The tweet is classified as positive by Rnn.
```

Figure 25 : Test sur un Tweet Positive Simple par Les modèles de DL.

```
# Example usage:
tweet_to_predict = "I like apple"
ML_predict_tweet_sentiment(tweet_to_predict)
```

```
The tweet is classified as positive by Decision Tree.
The tweet is classified as positive by Logistic Regression.
The tweet is classified as positive by Naive Bayes.
The tweet is classified as positive by SGD Regression.
The tweet is classified as positive by SVM.
The tweet is classified as positive by Random Forest Classifier.
The tweet is classified as positive by KNN.
```

Figure 26 : Test sur un Tweet Positive Simple par Les modèles de ML.

On remarque que les modèles de ML pour un tweet simple ont pu prédire la classe correctement, ce qui est dû à la simplicité du tweet.

5. Analyse des Travaux de Classification des Tweets

Dans ce qui suit, nous allons présenter quelques travaux qui utilisent les techniques de l'apprentissage automatique de manières diverses pour classifier les tweets :

- [Parveen *et al*., 21] les auteurs proposent une approche novatrice pour l'analyse des sentiments sur Twitter en utilisant une architecture de réseau récurrent à attention hybride (GARN). L'étude se concentre sur la classification des étiquettes de sentiment (positif, négatif, neutre) en prétraitant d'abord l'ensemble de données Sentiment 140, puis en extrayant des caractéristiques à l'aide du modèle de fréquence modifiée de terme de longueur (LTF-MICF). Un optimiseur hybride basé sur une mutation (HMWSO) est introduit pour la sélection des caractéristiques, suivi de la classification des sentiments en utilisant l'architecture GARN, qui combine des réseaux neuronaux récurrents (RNN) et des mécanismes d'attention. Le modèle GARN proposé atteint des métriques de performance élevées, notamment une précision (96,65%), un rappel (96,76%) et une mesure F (96,70%), démontrant son efficacité dans l'amélioration de la précision de l'analyse des sentiments et de l'efficacité du système sur les données Twitter.

- [Hilmiaji *et al*., 21] l'article de recherche se concentre sur l'identification des émotions dans les tweets indonésiens en utilisant un Réseau de Neurones Convolutif (CNN). L'approche implique de transformer les tweets en une séquence d'entiers et d'apprendre les structures linguistiques grâce à une couche d'incrustation. Le modèle classifie les tweets en fonction de leur structure linguistique en indonésien, en utilisant des paramètres tels que le taux d'apprentissage, le dropout et l'optimiseur pour l'expérimentation. La méthode de prétraitement implique de tokeniser les tweets, d'ajouter un rembourrage de zéros pour correspondre aux longueurs des tweets, et de construire un index d'incrustations à partir du dictionnaire GloVe. Le modèle de classification a atteint une précision, un rappel et un score F1 de 90,1%, 90,3% et 90,2% respectivement, avec la précision la plus élevée atteignant 89,8%. Ces résultats démontrent l'efficacité du modèle CNN dans l'identification précise des émotions dans les tweets indonésiens.

- [Olusegun *et al*., 23] les auteurs ont utilisé des modèles d'apprentissage en profondeur tels que CNN, LSTM, BiLSTM et CLSTM pour la classification des émotions, en abordant le déséquilibre des classes avec des techniques SMOTE et de sous-échantillonnage aléatoire. Le prétraitement impliquait l'extraction et l'analyse de 800 000 ensembles de données en utilisant le lexique NRCLexicon pour la prédiction de la signification émotionnelle. Le modèle CNN a surpassé les autres avec une précision remarquable de 96%. LSTM, bien que moins performant, a tout de même atteint une précision élevée de 94%. Les valeurs de précision, de rappel et de score F1 pour chaque modèle n'ont pas été explicitement fournies dans les contextes donnés. Dans l'ensemble, l'approche de l'étude combinant l'apprentissage en profondeur avec les techniques de traitement automatique du langage naturel, le prétraitement des ensembles de données et l'évaluation des modèles ont permis d'obtenir des informations précieuses sur les émotions publiques lors de l'épidémie de monkeypox, aidant potentiellement aux interventions en santé publique et à la compréhension des maladies.

Conclusion

Dans cette étude, nous avons entrepris une exploration approfondie des techniques de classification de sentiments sur les réseaux sociaux en utilisant à la fois des méthodes de machine learning traditionnelles et des modèles de deep learning. Nous avons établi une méthodologie rigoureuse pour prétraiter les données, construire des caractéristiques pertinentes et entraîner différents algorithmes de classification. Les résultats obtenus ont mis en lumière l'efficacité des modèles de deep learning, tels que les réseaux de neurones et les architectures récurrentes, pour capturer la complexité des données textuelles et prédire avec précision les sentiments des tweets. Leur capacité à apprendre des représentations de haute qualité à partir des données brutes a permis d'atteindre des performances remarquables, surpassant souvent les modèles de machine learning traditionnels. Ces résultats soulignent l'importance croissante des approches basées sur le deep learning dans le domaine de l'analyse des sentiments de tweets.

Bibliographie

[Parveen *et al.*, 21] Parveen, N., Chakrabarti, P., Hung, B., Shaik, A. (2021). "Twitter sentiment analysis using hybrid gated attention recurrent network. Journal of Sentiment Analysis", 1(1), 1-10.

[Hilmiaji *et al.*, 21] Hilmiaji, N., Lhaksmana, K., & Purbolaksono, M. (2021). Identifying Emotion on Indonesian Tweets using Convolutional Neural Networks. Journal of Natural Language Processing.

[Olusegun *et al.*, 23] Olusegun, R., Timothy O., Halima, A., Yao H. & Staphord, B. "Text Mining and Emotion Classification on Monkeypox Twitter Dataset: A Deep Learning-Natural Language Processing (NLP) Approach." Department of Computer Science, Bowie State University, Bowie, MD 20715, USA, and Department of Computer Science, Morgan State University, Baltimore, MD 21251, USA, 2023.