Université Mohammed V - Rabat
École Nationale d'Informatique
et d'Analyse des Systèmes

جامعة محمد الخامس بالرباط
Université Mohammed V de Rabat

Rapport de projet text mining

Filière

Genie Data

Sujet :

---

prédiction des maladies à partir des données médicaux

Réalisé par :

ARBAOUI Hamza

EL KAOUTHAR Mohamed

Encadré par :

M. TABII Youness

Année Universitaire 2024-2025

Table de matières

I - Introduction générale

L'analyse automatique de documents médicaux pour la prédiction de maladies représente une avancée majeure dans le domaine des technologies de la santé. En effet, la quantité massive de données textuelles générées quotidiennement par les hôpitaux, les cabinets médicaux, et les publications scientifiques contient des informations cliniques cruciales. Ces données, lorsqu'elles sont correctement exploitées, peuvent non seulement aider les professionnels de santé à prendre des décisions diagnostiques, mais aussi à identifier des tendances de santé publique, anticiper les risques de maladies, et améliorer les parcours de soins des patients. L'analyse de ces documents médicaux pour la prédiction de maladies s'inscrit donc dans une dynamique d'innovation visant à intégrer le traitement du langage naturel (NLP) au service de la médecine et de la santé publique.

L'objectif de ce projet est de développer une approche de prédiction de maladies à partir de documents médicaux en s'appuyant sur des techniques avancées de machine learning et de NLP. Plus précisément, ce travail vise à classer automatiquement les textes médicaux en fonction de différentes catégories de maladies (par exemple, les néoplasmes, les maladies du système digestif, les affections du système nerveux, et les maladies cardiovasculaires). En utilisant des modèles de NLP adaptés au contexte médical, nous espérons démontrer la capacité de ces techniques à comprendre des textes spécialisés et à fournir des prédictions précises et fiables.

Dans cette optique, ce rapport explore les différentes étapes de la chaîne de traitement, de l'acquisition et de la prétraitement des données, à la sélection et l'optimisation des modèles de classification. Le dataset utilisé pour cette étude comprend des documents médicaux annotés en fonction des catégories de maladies, ce qui permet de s'assurer que le modèle peut apprendre à différencier les termes spécifiques à chaque type de maladie. Nous utilisons des modèles de classification supervisée, intégrant des algorithmes de pointe dans le domaine du NLP, tels que BioBERT et DeBERTa, qui ont prouvé leur efficacité dans des tâches de compréhension de texte médical.

Les résultats de ce travail pourraient contribuer à l'amélioration des outils de diagnostic automatisé et à l'enrichissement des bases de données de recherche médicale, facilitant ainsi la mise en œuvre d'applications concrètes dans les systèmes de santé. En fin de compte, ce projet illustre comment les technologies de traitement automatique des langues peuvent jouer un rôle clé dans la révolution des pratiques médicales, en ouvrant de nouvelles perspectives pour une médecine plus prédictive, préventive, et personnalisée.

II - Exploration des données

2-1 Description dataset

Le dataset utilisé dans ce projet contient des résumés médicaux (ou abstracts) décrivant les états cliniques de patients. Ce type de données est couramment utilisé par les professionnels de la santé, en particulier les médecins, pour analyser rapidement les informations pertinentes et déterminer les affections dont souffrent les patients. Ce projet vise à concevoir une technologie d'assistance capable d'identifier avec précision la classe de problèmes médicaux décrite dans chaque résumé.

Le dataset est divisé en deux ensembles de données distincts :

2-1-1 Ensemble d'entraînement :

- Nombre d'enregistrements : 14 438 résumés médicaux.
- Labels: Les résumés de l'ensemble d'entraînement sont étiquetés selon cinq classes de maladies :
  - **Digestive System Diseases** – Maladies affectant le système digestif.
  - **Cardiovascular Diseases** – Pathologies du système cardiovasculaire, comme les maladies coronariennes.
  - **Neoplasms** – Affections tumorales, incluant les cancers et les tumeurs bénignes.
  - **Nervous System Diseases** – Troubles du système nerveux, tels que les maladies neurodégénératives.
  - **General Pathological Conditions** – Autres conditions pathologiques d'ordre général, sans lien spécifique avec un organe particulier.
- Format des données : Les données sont fournies dans un fichier texte (train.dat) contenant les abstracts, chacun associé à une des cinq classes

mentionnées.

2-1-2 Ensemble de test :

- **Nombre d'enregistrements** : 14 442 résumés médicaux non étiquetés.
- **Objectif** : Les classes de maladies ne sont pas fournies pour cet ensemble de données. L'objectif du modèle est de prédire les étiquettes pour chacun des abstracts présents dans cet ensemble.
- **Format des données** : Les abstracts sont fournis dans un fichier texte (test.dat) qui doit être prétraité et classifié.

	label	texte
0	4	Catheterization laboratory events and hospital...
1	5	Renal abscess in children. Three cases of rena...
2	2	Hyperplastic polyps seen at sigmoidoscopy are ...
3	5	Subclavian artery to innominate vein fistula a...
4	4	Effect of local inhibition of gamma-aminobutyr...
...	...	...
14433	4	Quadricuspid aortic valve and aortic regurgita...
14434	1	Mammographic measurements before and after aug...
14435	1	Use of leukocyte-depleted platelet concentrate...
14436	2	Complications of Tenckhoff catheters post remo...
14437	3	Fatal or severely disabling cerebral infarctio...

14438 rows × 2 columns

-

2-1-3 Structure des Données :

Chaque document du dataset est structuré sous forme de texte brut décrivant un état clinique spécifique. Ces résumés incluent des informations sur les symptômes, diagnostics, traitements, et autres observations médicales importantes

Les documents contiennent des termes médicaux et techniques adaptés à chaque catégorie de maladie, ce qui en fait un ensemble de données spécialisé pour les modèles de traitement du langage naturel dans le domaine médical.

Ce dataset offre une base riche pour développer et tester des modèles de classification de textes médicaux, avec pour but ultime de faciliter

l'identification automatique des affections à partir de descriptions cliniques.

2-2 Visualisation des données

La visualisation des données textuelles est essentielle pour comprendre les caractéristiques linguistiques propres à chaque catégorie de maladies présentes dans le dataset. En analysant la fréquence des mots dans les abstracts médicaux pour chaque classe de maladies, il est possible d'identifier les termes les plus représentatifs de chaque catégorie. Ces visualisations, sous forme de nuages de mots et de graphiques en barres, permettent de mettre en évidence les différences terminologiques entre les classes, offrant un aperçu des mots clés les plus fréquents dans chaque domaine médical.

2-2-1 Exploration de la Fréquence des Mots par Classe

Pour chaque classe de maladies, nous avons extrait les dix mots les plus fréquents. Les classes incluent les **maladies du système digestif**, les **maladies cardiovasculaires**, les **néoplasies** (tumeurs et cancers), les **maladies du système nerveux**, et les **conditions pathologiques générales**. Cette analyse est réalisée en excluant les mots courants de la langue anglaise pour se concentrer uniquement sur les termes médicaux pertinents.

Les mots les plus fréquents de chaque classe révèlent des caractéristiques linguistiques spécifiques. Par exemple, pour les maladies cardiovasculaires, des mots comme "cardiaque" ou "coronaire" apparaissent souvent, tandis que dans les maladies du système nerveux, des termes comme "neuro" ou "cérébral" sont prédominants. Ces mots constituent un point d'appui pour l’entraînement des modèles de NLP, car ils aident le modèle à mieux distinguer les catégories de maladies.

2-2-2 Visualisation par Nuages de Mots

Les **nuages de mots** permettent de représenter visuellement les termes les plus fréquents de chaque classe. Plus un mot apparaît fréquemment dans une catégorie, plus il est affiché en grand dans le nuage. Ces visualisations sont utiles pour percevoir les termes importants d’un coup d'œil, et elles montrent les mots qui se distinguent dans chaque catégorie. En insérant un nuage de mots pour chaque classe, nous pouvons rapidement comprendre les thématiques linguistiques spécifiques aux différentes maladies.

FIG: Mot les plus fréquents dans la classe 4

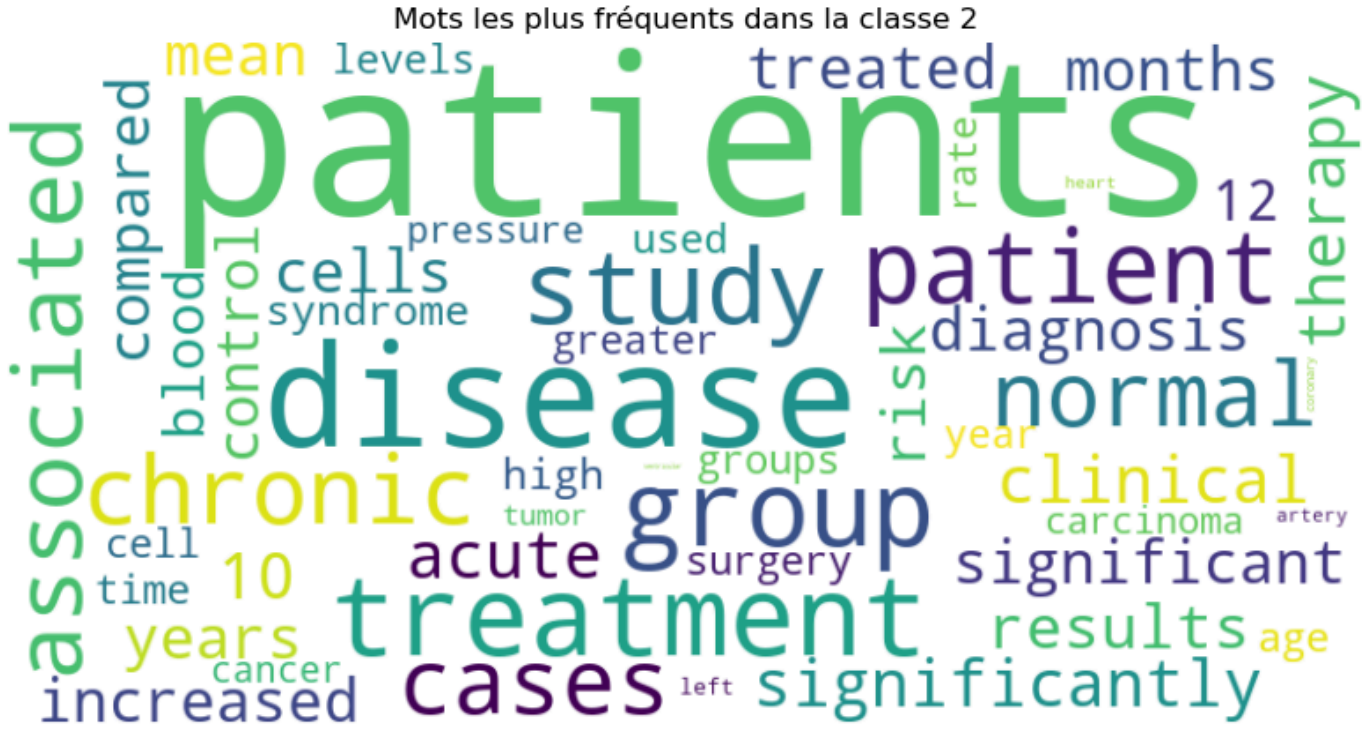


FIG: Mot les plus fréquents dans la classe 2

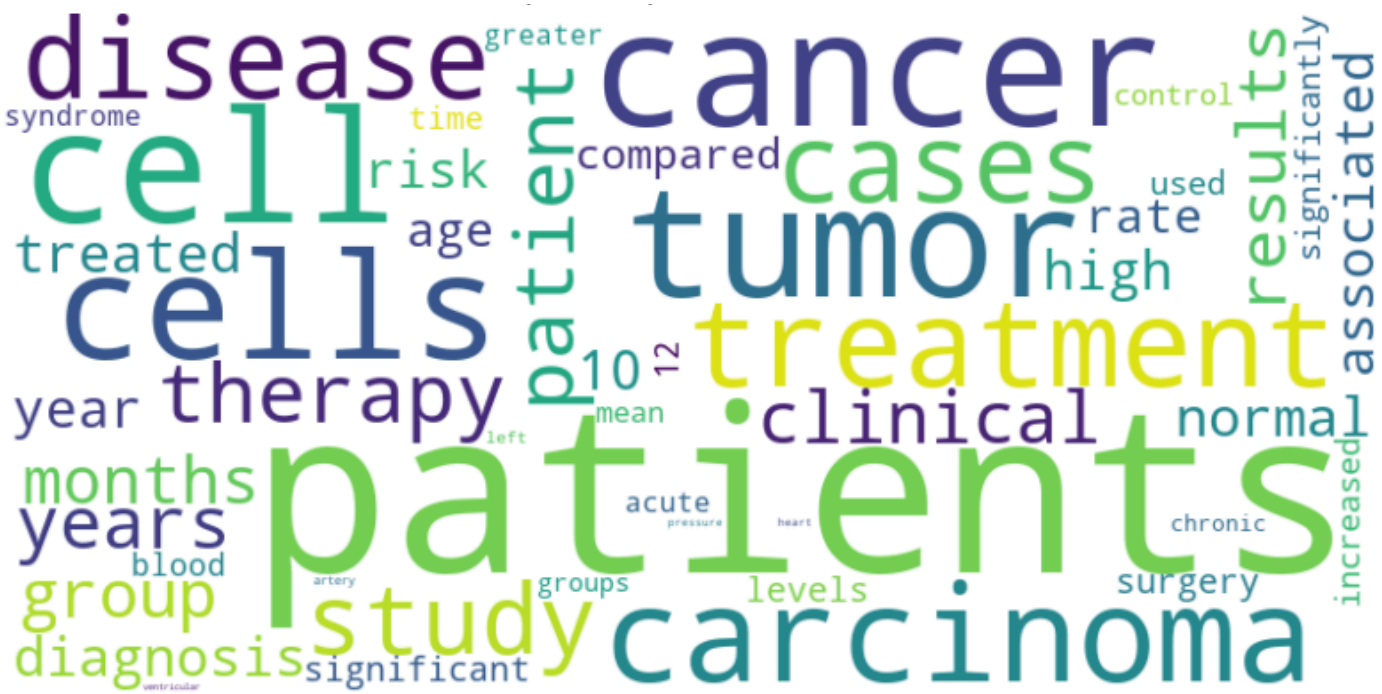


FIG: Mot les plus fréquents dans la classe 1

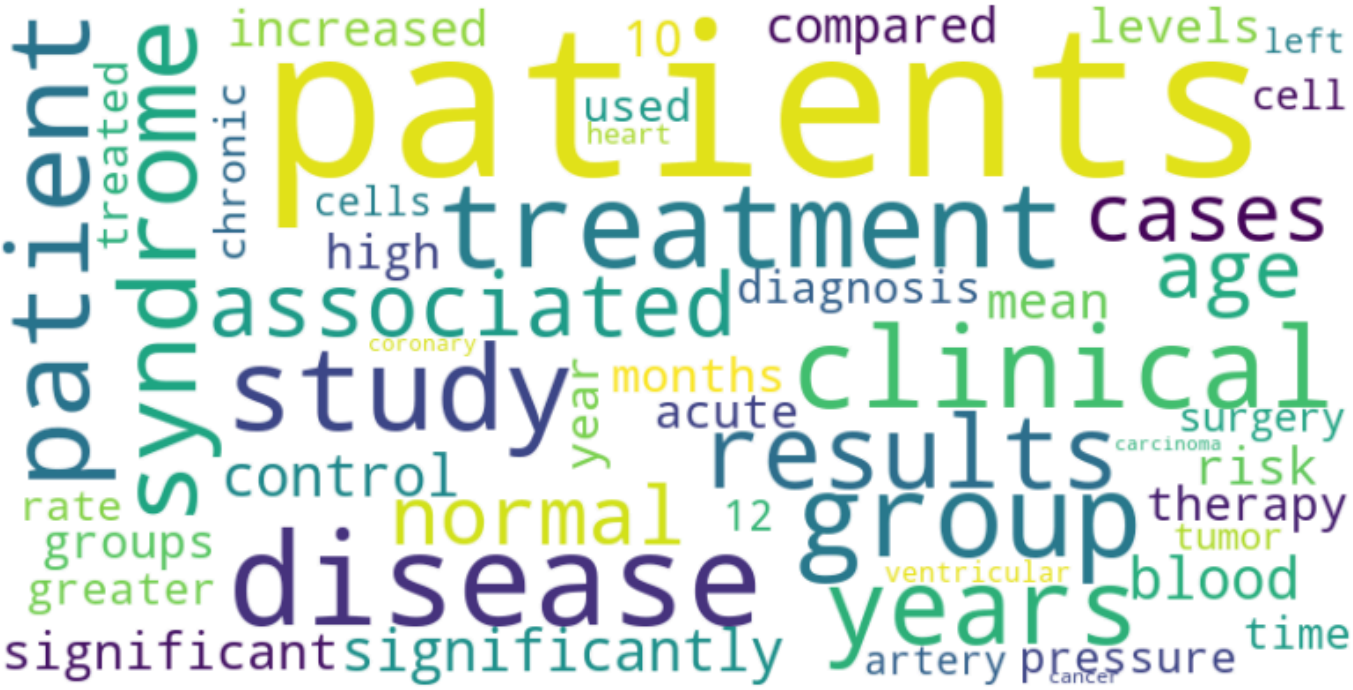


FIG: Mot les plus fréquents dans la classe 3

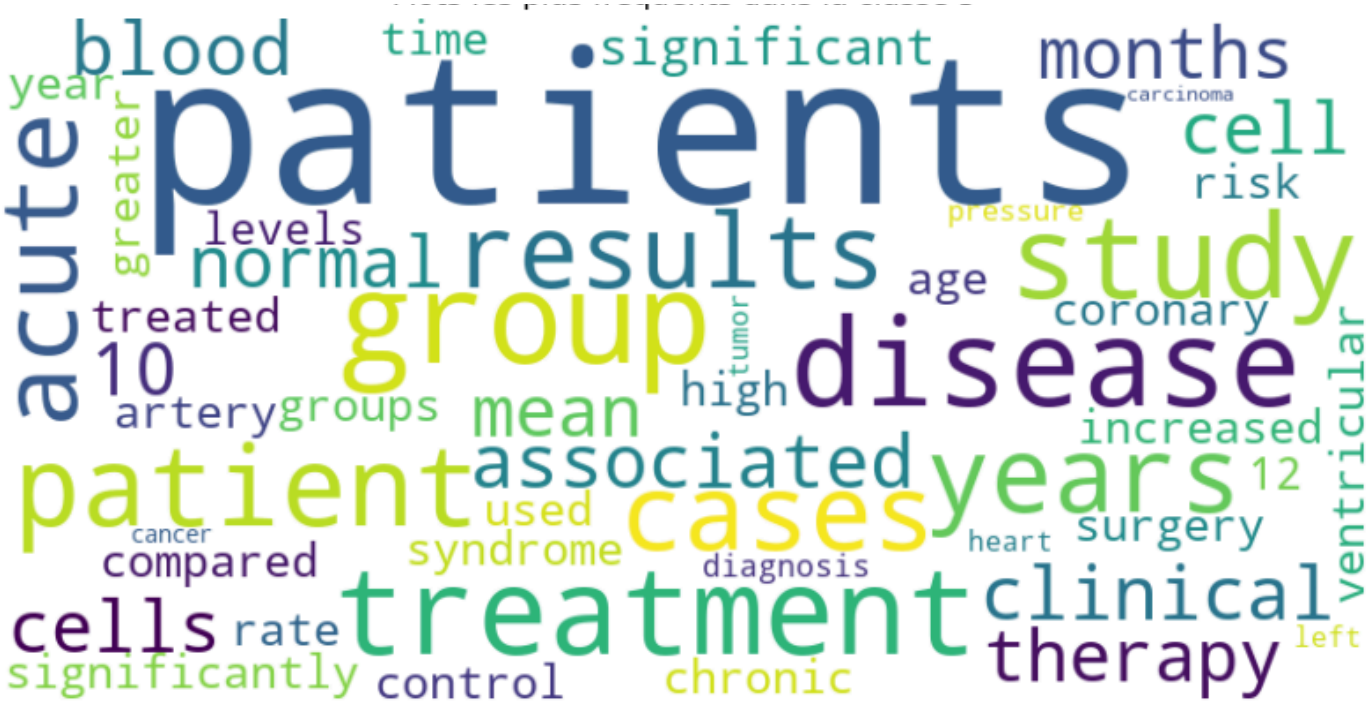
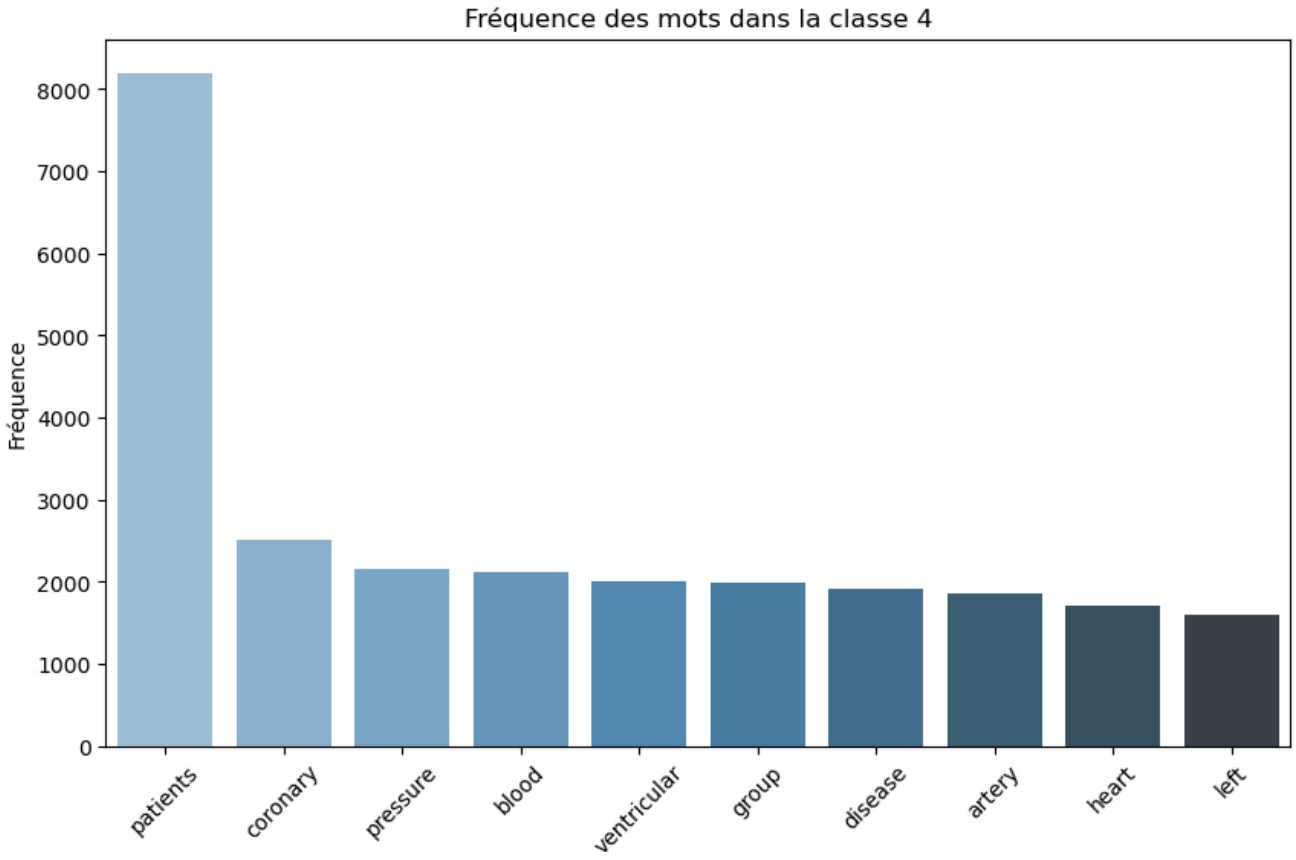
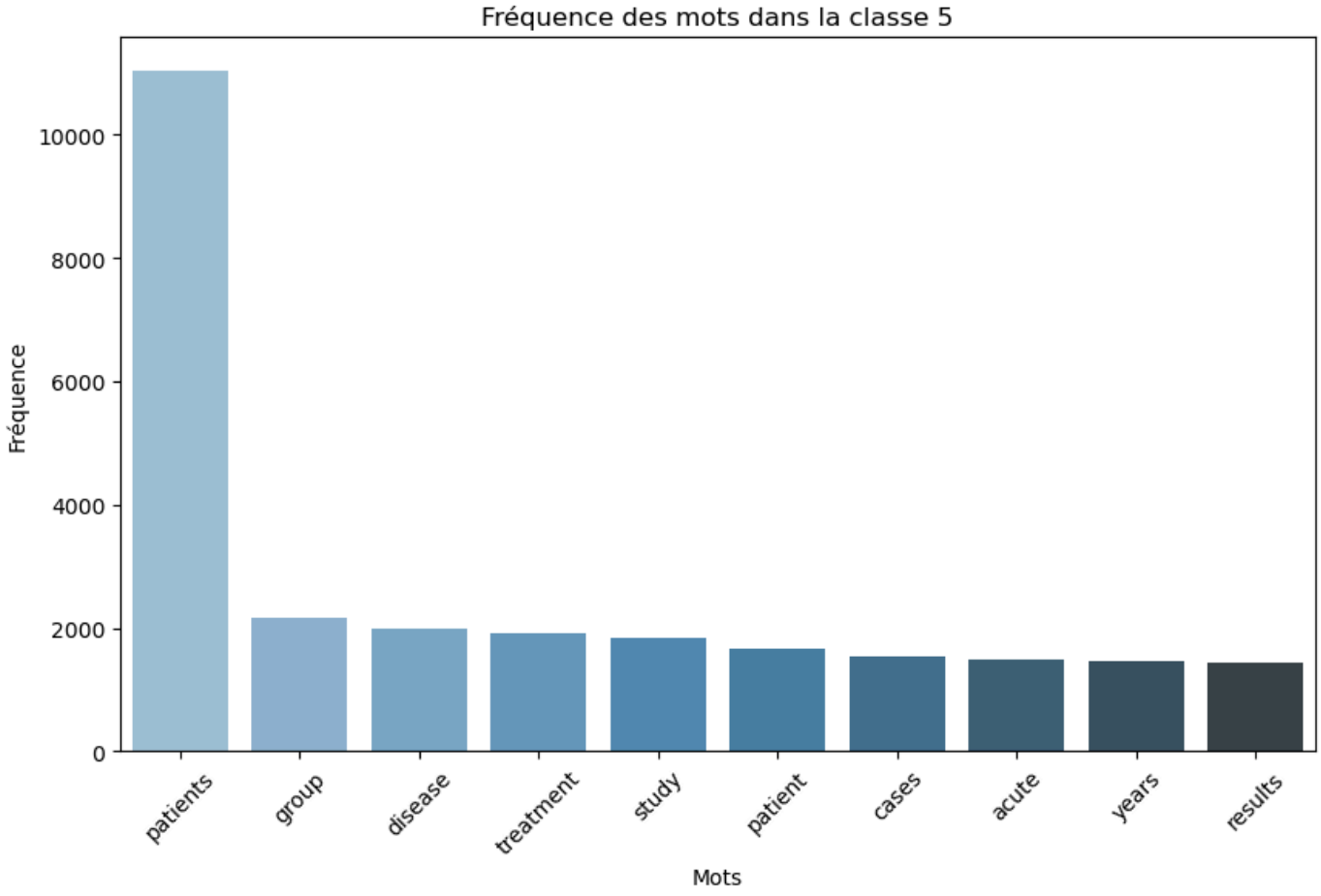


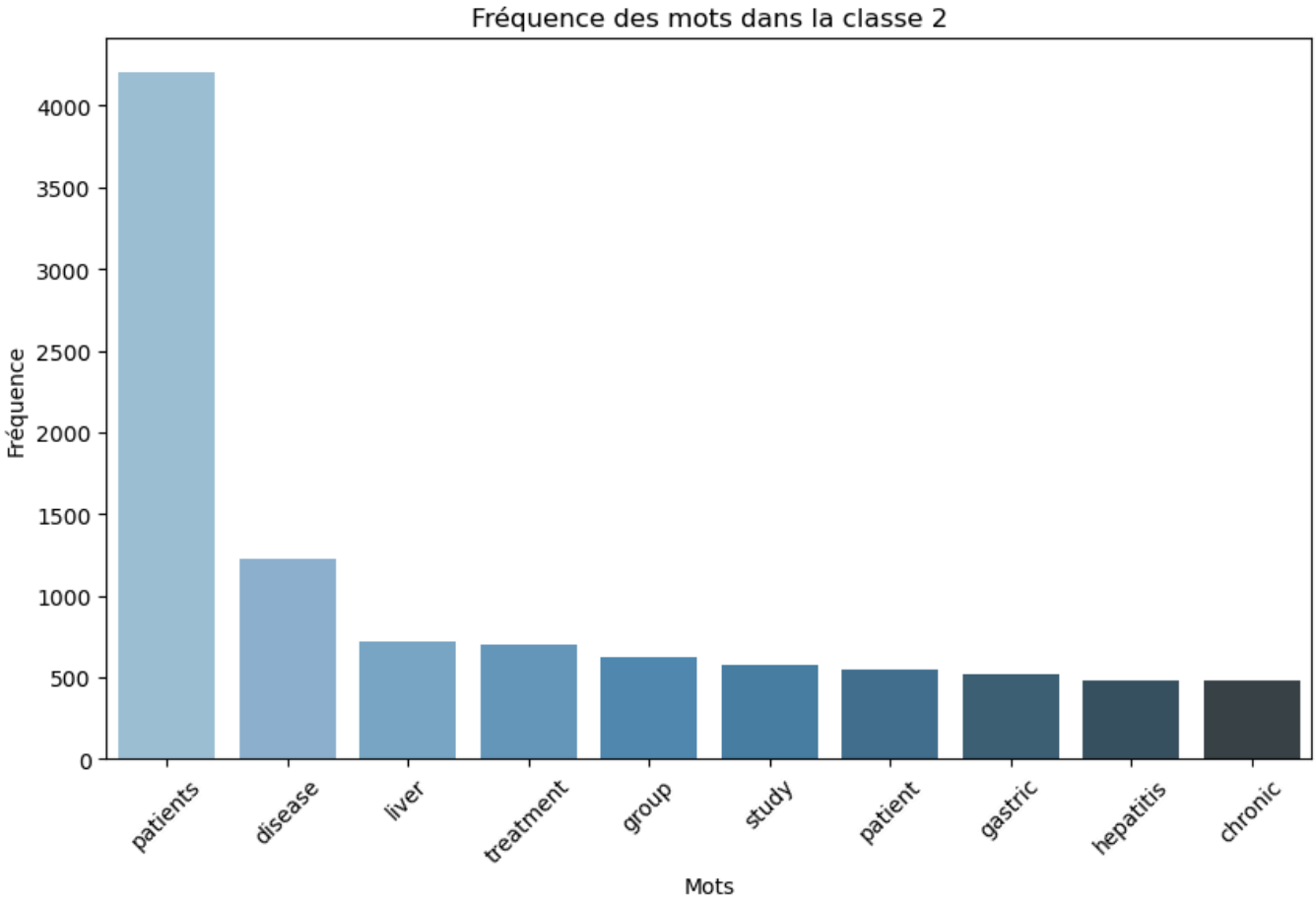
FIG: Mot les plus fréquents dans la classe 5

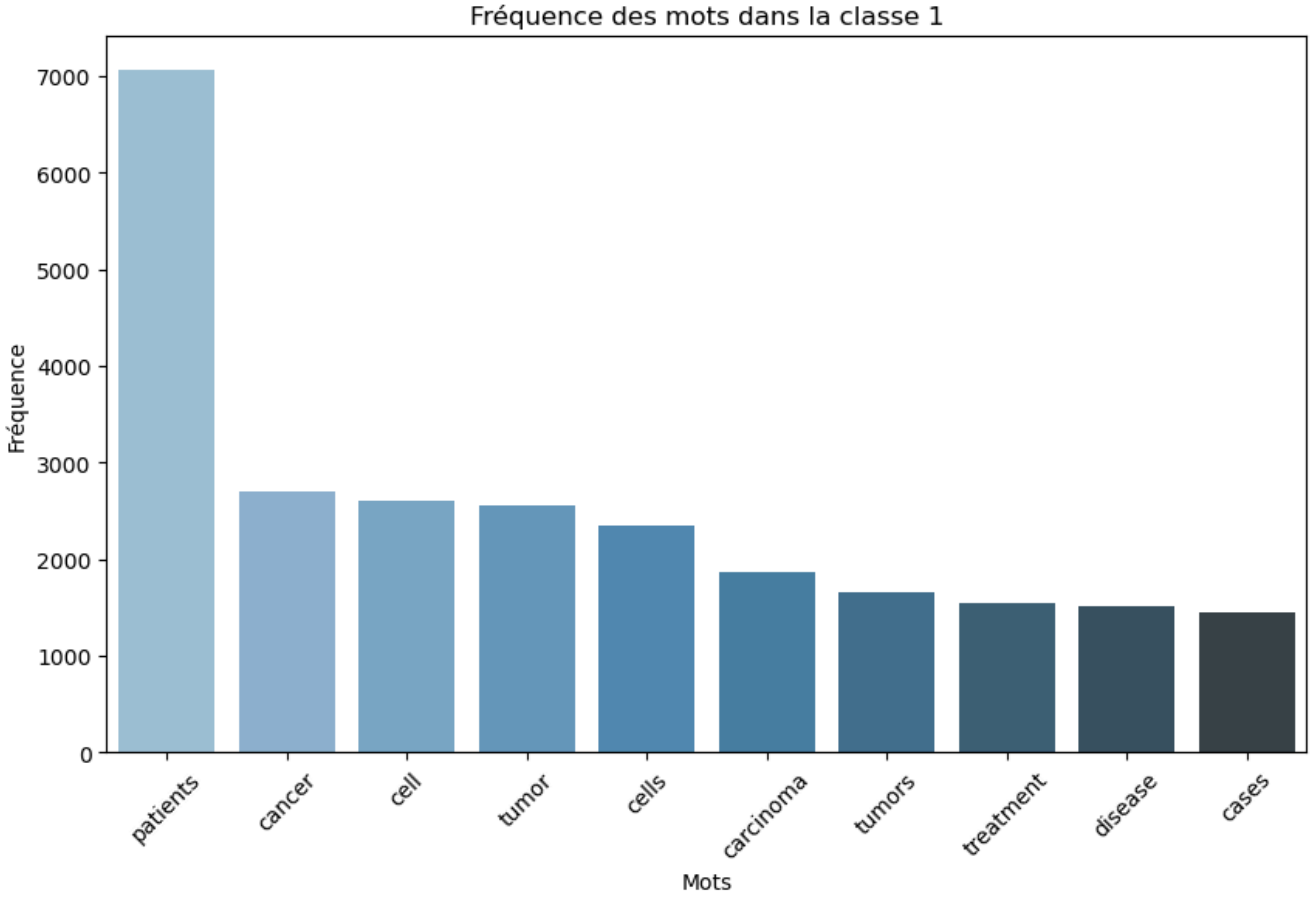
2-2-3 Visualisation par Graphiques en Barres

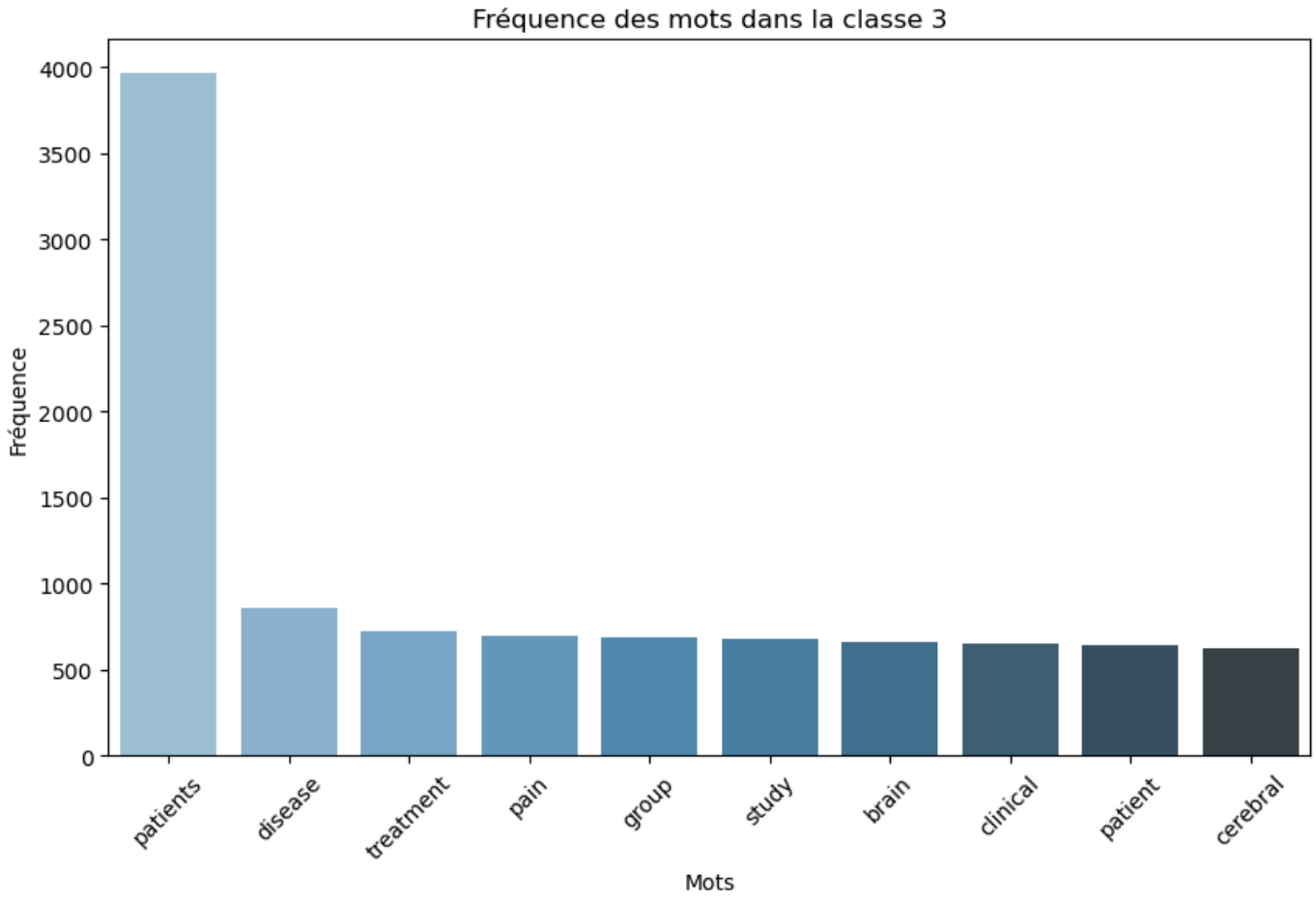
Les **graphes en barres** complètent les nuages de mots en offrant une représentation quantitative des dix mots les plus fréquents pour chaque classe. Cette visualisation permet de comparer les fréquences des mots et de mieux comprendre leur importance relative au sein de chaque catégorie. Pour chaque classe de maladies, le graphe en barres montre la fréquence des mots les plus représentatifs, ce qui fournit une vue plus précise de la répartition des termes médicaux dans le dataset.


Fréquence des mots dans la classe 4
Fréquence
0
1000
2000
3000
4000
5000
6000
7000
8000
patients
coronary
pressure
blood
ventricular
group
disease
artery
heart
left


Fréquence des mots dans la classe 5
Fréquence
0
2000
4000
6000
8000
10000
patients
group
disease
treatment
study
patient
cases
acute
years
results
Mots


Fréquence des mots dans la classe 2
Fréquence
0
500
1000
1500
2000
2500
3000
3500
4000
patients
disease
liver
treatment
group
study
patient
gastric
hepatitis
chronic
Mots


Fréquence des mots dans la classe 1
Fréquence
0
1000
2000
3000
4000
5000
6000
7000
patients
cancer
cell
tumor
cells
carcinoma
tumors
treatment
disease
cases
Mots



2-2-4 Interprétation des Résultats

Ces visualisations mettent en lumière les différences linguistiques entre les catégories de maladies. Les mots-clés identifiés permettent de cerner les caractéristiques terminologiques propres à chaque classe, ce qui renforce la compréhension de l'ensemble des données. Par ailleurs, cette analyse facilite le choix des prétraitements et des méthodes de sélection des caractéristiques pour l'entraînement des modèles de NLP. Les insights obtenus à partir de ces visualisations contribuent à la précision et à la robustesse des modèles de classification de maladies.

2-3 Prétraitement des Données Textuelles

Le prétraitement des données textuelles est une étape cruciale dans la préparation des documents médicaux avant l'entraînement des modèles de classification. Les textes médicaux contiennent souvent des termes techniques, des abréviations, et des informations superflues, rendant essentiel un processus de nettoyage et de normalisation. Ce sous-chapitre présente les étapes de prétraitement appliquées au dataset pour transformer les documents

bruts en données prêtes à être exploitées par les algorithmes de machine learning.

2-3-1. Normalisation des Documents Textuels

La fonction de normalisation utilisée, normalize_document, applique une série d'opérations visant à uniformiser et simplifier les documents. Ce processus inclut les étapes suivantes :

- Conversion en minuscules et suppression des caractères spéciaux : Les documents sont convertis en minuscules pour éviter la distinction entre les mots identiques avec des majuscules. Tous les caractères non alphabétiques sont supprimés, ce qui élimine la ponctuation et les symboles spéciaux, sources de bruit dans les modèles.
- Suppression des espaces blancs : Les espaces superflus sont enlevés pour garantir une structure textuelle homogène.
- Tokenisation des mots : Le texte est divisé en tokens (mots) individuels pour permettre des opérations de traitement mot par mot.
- Suppression des stopwords : Les mots les plus courants, ou “stopwords” (par exemple, "and", "the", "is"), sont retirés, car ils n'apportent généralement pas d’information spécifique à la classification.
- Racisation (Stemming) : Chaque mot est réduit à sa racine par un processus de stemming à l'aide de l’algorithme PorterStemmer. Cette étape permet de regrouper les variantes d’un même mot, ce qui facilite la reconnaissance des concepts similaires (par exemple, "diagnose" et "diagnosing" deviennent "diagnos").

L’ensemble de ces opérations est appliqué à chaque document pour créer une version normalisée, prête à être transformée en vecteurs numériques.

2-3-2. Vectorisation TF-IDF

Pour représenter les documents textuels en tant que vecteurs, nous avons utilisé la vectorisation TF-IDF (Term Frequency-Inverse Document Frequency). Cette technique permet d'évaluer l'importance d’un mot dans un document par rapport à l'ensemble du corpus. Les étapes sont les suivantes :

- Calcul des fréquences de termes (TF) : Le nombre de fois qu’un mot apparaît dans un document est calculé, offrant une première mesure de son importance dans ce document.
- Pondération par fréquence inverse de document (IDF) : La fréquence d'apparition d’un mot dans l'ensemble des documents est prise en compte

pour réduire le poids des mots trop communs et valoriser ceux qui sont plus spécifiques.
- Transformation en vecteurs TF-IDF : La fonction TfidfVectorizer de sklearn est appliquée aux documents normalisés pour générer une matrice TF-IDF. Chaque document est ainsi représenté par un vecteur numérique, où chaque valeur représente le poids d'un mot spécifique dans ce document.

2-3-3. Résultat du Prétraitement

Après le prétraitement, chaque document est représenté par un vecteur de poids TF-IDF, stocké dans une matrice prête à être utilisée par les modèles de classification. La dimensionnalité des documents est ainsi réduite, permettant de concentrer les modèles sur les informations lexicales les plus pertinentes pour la prédiction des classes de maladies.

Ce prétraitement permet non seulement de simplifier les données textuelles, mais aussi d'extraire les informations essentielles propres à chaque classe de maladie, facilitant ainsi la tâche des modèles de machine learning dans l’identification des pathologies décrites.

III. Choix de modèle

3.1 Présentation des modèles

3.1.a Random forest

Le modèle Random Forest est une méthode d'ensemble qui combine plusieurs arbres de décision pour améliorer la précision et réduire le risque de surapprentissage (overfitting). Chaque arbre de la forêt est construit en utilisant un sous-échantillon aléatoire des données et une sélection aléatoire de caractéristiques à chaque nœud, ce qui renforce la robustesse et la généralisation du modèle. La prédiction finale est généralement obtenue par un vote majoritaire dans les tâches de classification, ou par une moyenne dans les tâches de régression.

Applications :
Random Forest est largement utilisé pour des tâches de classification et de régression dans divers domaines, tels que la finance, la médecine, et les

systèmes de recommandation. Sa capacité à traiter des données de grande dimension et à fournir des prédictions robustes le rend très polyvalent.

Forces et faiblesses :

- ***Forces* :** Random Forest est robuste face aux valeurs aberrantes et au surapprentissage, et il peut traiter de nombreuses caractéristiques sans nécessiter une normalisation importante. Le modèle offre aussi une importance des variables, ce qui peut être utile pour interpréter les résultats.
- ***Faiblesses* :** Il est plus complexe et plus coûteux en temps de calcul que les modèles d'arbres individuels, surtout pour des grands ensembles de données. Les prédictions peuvent également manquer d'interprétabilité car elles reposent sur une combinaison de nombreux arbres.

Pourquoi utiliser Random Forest dans ce projet :
Random Forest est un modèle puissant et fiable qui peut améliorer la précision par rapport aux modèles individuels comme l'arbre de décision. Dans ce projet, il peut offrir des résultats stables et robustes pour la classification de texte en s'appuyant sur la diversité des arbres pour mieux capturer la complexité des données.

3.1.b K-Nearest Neighbors (KNN)

Le modèle des K plus proches voisins (KNN) est une méthode d'apprentissage supervisé utilisée principalement pour les tâches de classification et de régression. Il classe un point de données en fonction de la majorité des étiquettes de ses voisins les plus proches. La distance entre les points est généralement mesurée à l'aide de métriques comme la distance euclidienne ou la distance de Manhattan.

Applications :
KNN est souvent utilisé pour des tâches de classification de texte ou d'images, de reconnaissance de formes, et dans les systèmes de recommandation. Il est particulièrement adapté lorsque les relations entre les échantillons sont simples et non linéaires.

Forces et faiblesses :

- ***Forces* :** Facile à comprendre, à implémenter, et souvent performant pour les petites bases de données.

- ***Faiblesses*** **:** Inefficace sur les ensembles de données volumineux, car il nécessite un calcul de distance pour chaque point. KNN est également sensible aux valeurs aberrantes et aux données déséquilibrées.

Pourquoi utiliser KNN dans ce projet :
Ce modèle est intéressant pour explorer des approches simples mais efficaces dans la classification des textes. Il peut servir de modèle de référence pour comparer la performance avec d'autres modèles plus complexes.

3.1.c K-means

Le K-Means est un algorithme de clustering non supervisé qui regroupe les données en un nombre spécifique (K) de clusters. L'algorithme fonctionne en minimisant la distance intra-cluster et en maximisant la distance inter-cluster. Chaque point de données est associé au cluster dont le centroïde est le plus proche.

Applications :
K-Means est utilisé pour la segmentation de données, l'analyse de groupes d'utilisateurs ou de clients, et le regroupement de documents en fonction de leur similarité.

Forces et faiblesses :

- ***Forces*** **:** Efficace pour des grandes bases de données et facile à interpréter. C'est également un modèle rapide pour les grandes dimensions de données.
- ***Faiblesses*** **:** Nécessite de connaître le nombre de clusters (K) à l'avance et est sensible aux valeurs aberrantes. Peut également être influencé par les clusters initialement choisis.

Pourquoi utiliser K-Means dans ce projet :
K-Means est utile pour identifier des groupes thématiques dans des textes. Dans ce projet, il permet de regrouper les textes en classes potentielles, même lorsque les étiquettes ne sont pas disponibles.

3.1.d Arbres de décision

Un arbre de décision est un modèle d'apprentissage supervisé qui prend la forme d'une structure arborescente où chaque nœud interne représente une condition (ou test) sur un attribut, chaque branche représente un résultat de ce test, et chaque feuille représente une classe de décision (dans le cas de la classification). L'algorithme de l'arbre de décision construit un modèle en

segmentant de manière récursive les données en sous-groupes jusqu'à obtenir des feuilles homogènes ou jusqu'à atteindre un critère d'arrêt.

Applications :
Les arbres de décision sont couramment utilisés dans les systèmes de diagnostic, la classification de texte, la segmentation de clients, et les modèles de prédiction où l'interprétabilité est importante. Ils sont particulièrement adaptés aux ensembles de données où les relations entre les variables sont simples.

Forces et faiblesses :

- ***Forces* :** Les arbres de décision sont faciles à interpréter et à visualiser, ce qui les rend utiles pour des applications nécessitant des décisions transparentes. Ils gèrent bien les données avec des valeurs manquantes et ne nécessitent pas une préparation complexe des données.
- ***Faiblesses* :** Les arbres de décision sont sujets au surapprentissage, surtout sur des données complexes. De plus, ils sont sensibles aux petites variations dans les données, qui peuvent entraîner une structure d'arbre différente.

Pourquoi utiliser un Arbre de décision dans ce projet :
Dans ce projet, l'arbre de décision peut être utilisé comme modèle de référence pour la classification de texte. Sa simplicité d'interprétation permet de comprendre les critères de classification, et il fournit une bonne base pour évaluer les performances d'autres modèles plus complexes.

3.2 Implémentations et résultats de chaque modèle

3.2.1 Random forest

```python
# Modèle 1 : Random Forest
print("Modèle Random Forest")
rf_model = RandomForestClassifier(n_estimators=200, max_depth=42, random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)

# Évaluation du modèle Random Forest
accuracy_rf = accuracy_score(y_test, y_pred_rf)
print(f"Accuracy: {accuracy_rf * 100:.2f}%")
print("\nClassification Report (Random Forest):")
print(classification_report(y_test, y_pred_rf, target_names=class_names))
```

Ajustement des hyperparamètres les plus performant :

Pour optimiser le modèle Random Forest et améliorer son **accuracy**, nous avons ajusté les hyperparamètres clés suivants :

n_estimators = 200 : Ce paramètre représente le nombre d'arbres de décision dans la forêt. Nous avons choisi une valeur de 200 pour augmenter la diversité des arbres et donc la robustesse du modèle. En effet, un nombre plus élevé d'arbres permet de capturer davantage de variations dans les données, mais avec un compromis sur le temps d'entraînement. Après plusieurs itérations, ce réglage nous a semblé un bon compromis pour obtenir des performances stables sans alourdir excessivement le calcul.

max_depth = 42 : La profondeur maximale de chaque arbre contrôle jusqu'où le modèle peut segmenter les données avant de s'arrêter. En limitant cette profondeur à 42, nous avons cherché à éviter le surapprentissage tout en permettant aux arbres de capturer des relations significatives dans les données. Un modèle trop profond risquerait de s'adapter excessivement aux données d'entraînement, réduisant ainsi sa capacité de généralisation.

Résultats du modèle :

Modèle Random Forest
Accuracy: 45.19%

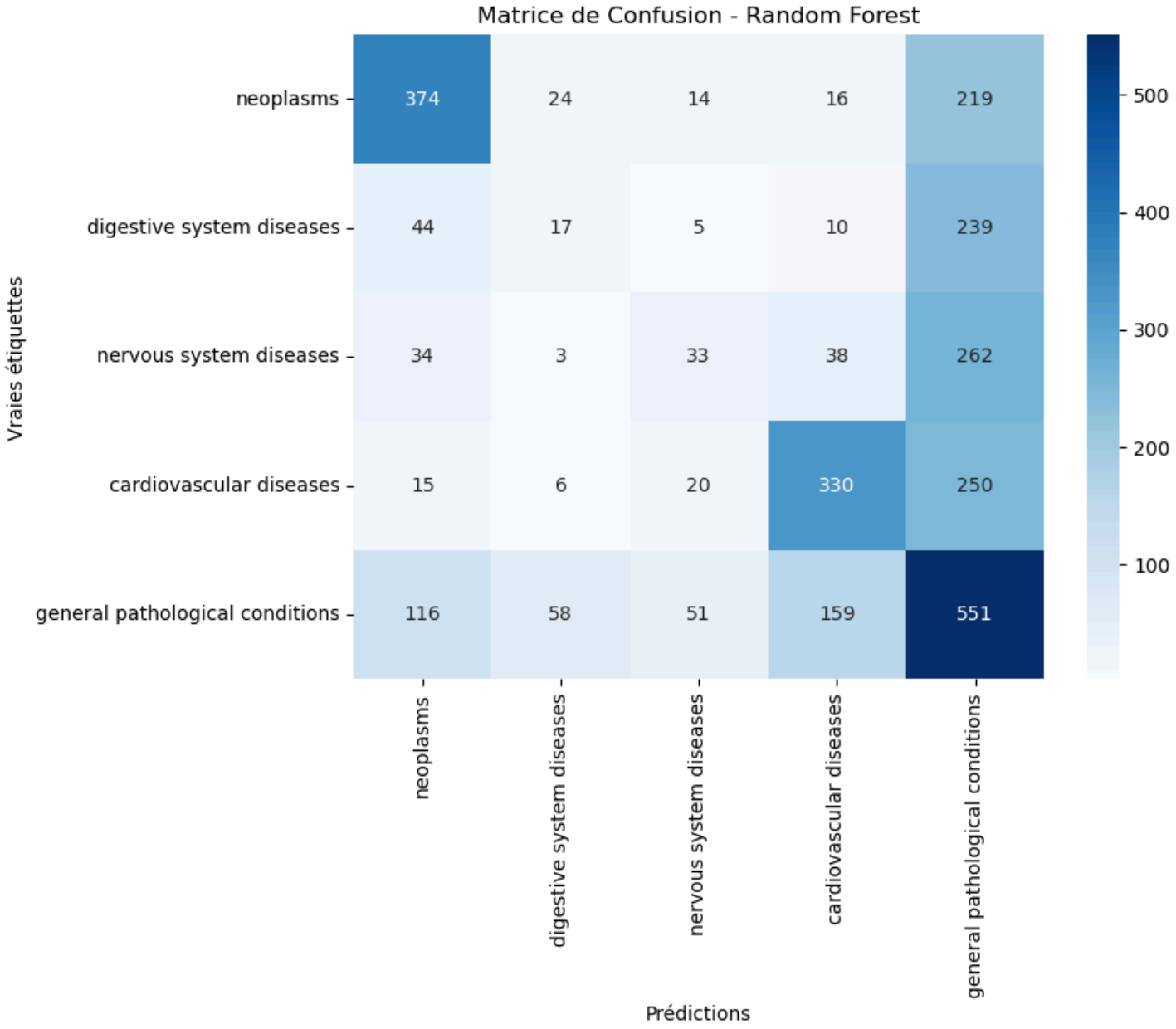
Classification Report (Random Forest):

	precision	recall	f1-score	support
neoplasms	0.64	0.58	0.61	647
digestive system diseases	0.16	0.05	0.08	315
nervous system diseases	0.27	0.09	0.13	370
cardiovascular diseases	0.60	0.53	0.56	621
general pathological conditions	0.36	0.59	0.45	935
accuracy			0.45	2888
macro avg	0.41	0.37	0.37	2888
weighted avg	0.44	0.45	0.43	2888

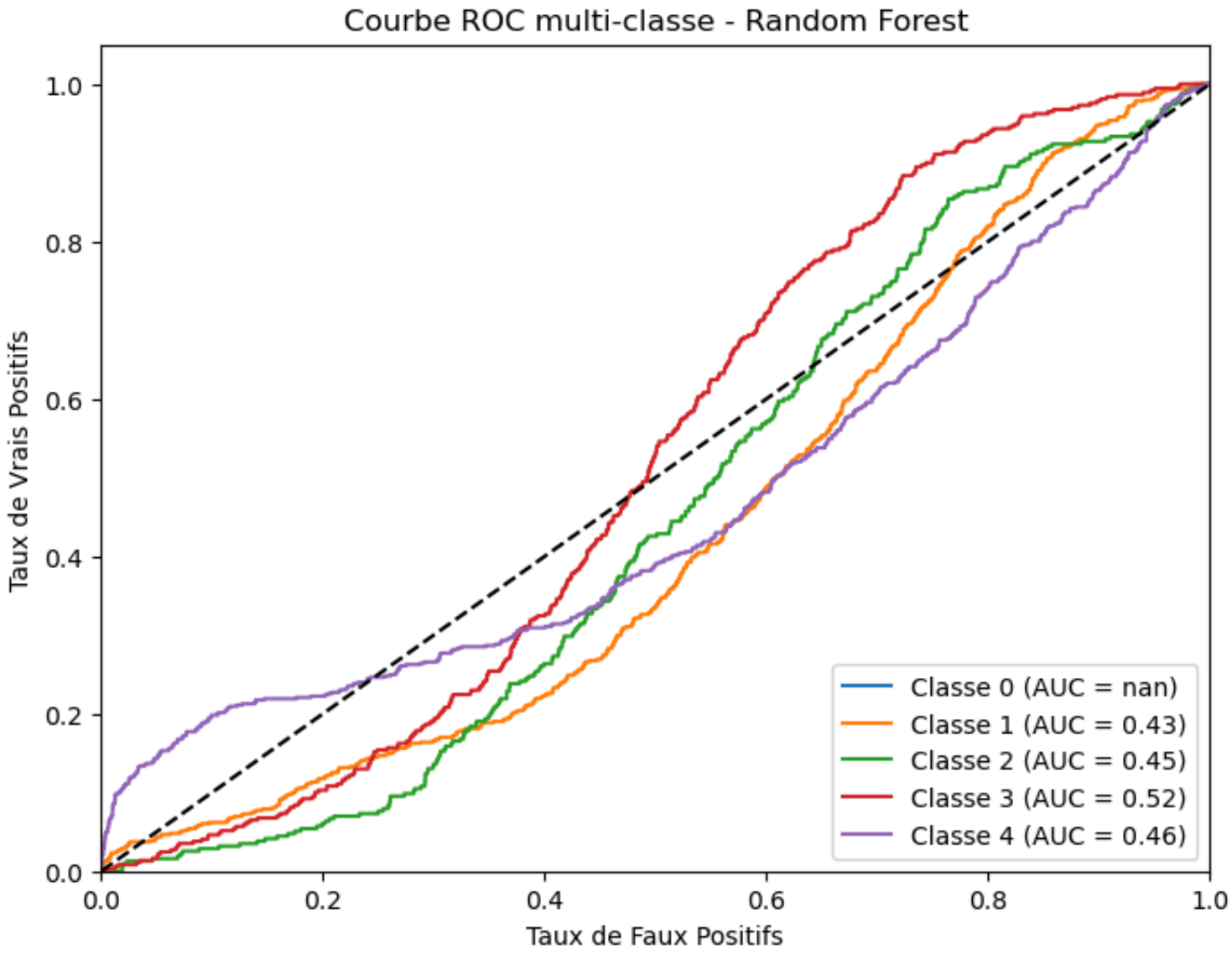
Résultat final :
Après optimisation, le modèle Random Forest a atteint une accuracy de **45.19%**. Bien que nous ayons ajusté ces hyperparamètres pour améliorer la performance, les résultats montrent que le modèle pourrait être limité par la nature des données elles-mêmes ou qu'il nécessiterait encore d'autres ajustements de paramètres pour une meilleure performance.

Rapport de classification :
Le modèle semble particulièrement bien fonctionner pour la classe **neoplasms** avec une précision de 0.64 et un rappel de 0.58, et la classe **cardiovascular diseases** avec des scores relativement plus élevés. Cependant, les classes telles que **digestive system diseases** et **nervous system diseases** ont montré des performances plus faibles, suggérant que le modèle pourrait avoir des difficultés à distinguer ces catégories avec la configuration actuelle des hyperparamètres.

Matrice de confusion :


Matrice de Confusion - Random Forest
neoplasms
digestive system diseases
nervous system diseases
cardiovascular diseases
general pathological conditions
Vraies étiquettes
374 24 14 16 219
44 17 5 10 239
34 3 33 38 262
15 6 20 330 250
116 58 51 159 551
neoplasms
digestive system diseases
nervous system diseases
cardiovascular diseases
general pathological conditions
Prédictions
500
400
300
200
100

Courbe ROC :



3.2.2 K-Nearest Neighbors (KNN)

```
# Modèle 2 : K-Nearest Neighbors (KNN)
print("\nModèle K-Nearest Neighbors (KNN)")
knn_model = KNeighborsClassifier(n_neighbors=77)  # Ajuster le nombre de voisins si nécessaire
knn_model.fit(X_train, y_train)
y_pred_knn = knn_model.predict(X_test)

# Évaluation du modèle KNN
accuracy_knn = accuracy_score(y_test, y_pred_knn)
print(f"Accuracy: {accuracy_knn * 100:.2f}%")
print("\nClassification Report (KNN):")
print(classification_report(y_test, y_pred_knn, target_names=class_names))
```

Ajustement des hyperparamètres les plus performant :

Pour optimiser le modèle KNN et améliorer son **accuracy**, nous avons ajusté les hyperparamètres clés suivants :

n_neighbors = 77 : Ce paramètre représente le nombre de voisins les plus proches utilisés pour déterminer la classe de chaque point. Une valeur élevée comme 77 signifie que le modèle utilise une plus grande quantité d'informations des voisins pour la classification, ce qui a tendance à stabiliser le modèle en réduisant l'impact des points aberrants. Après plusieurs essais, cette valeur de 77 a donné les meilleures performances en matière de précision globale, bien qu'un nombre plus bas de voisins puisse donner des résultats différents sur d'autres types de données.

Résultats du modèle :

```
Modèle K-Nearest Neighbors (KNN)
Accuracy: 62.02%

Classification Report (KNN):
```

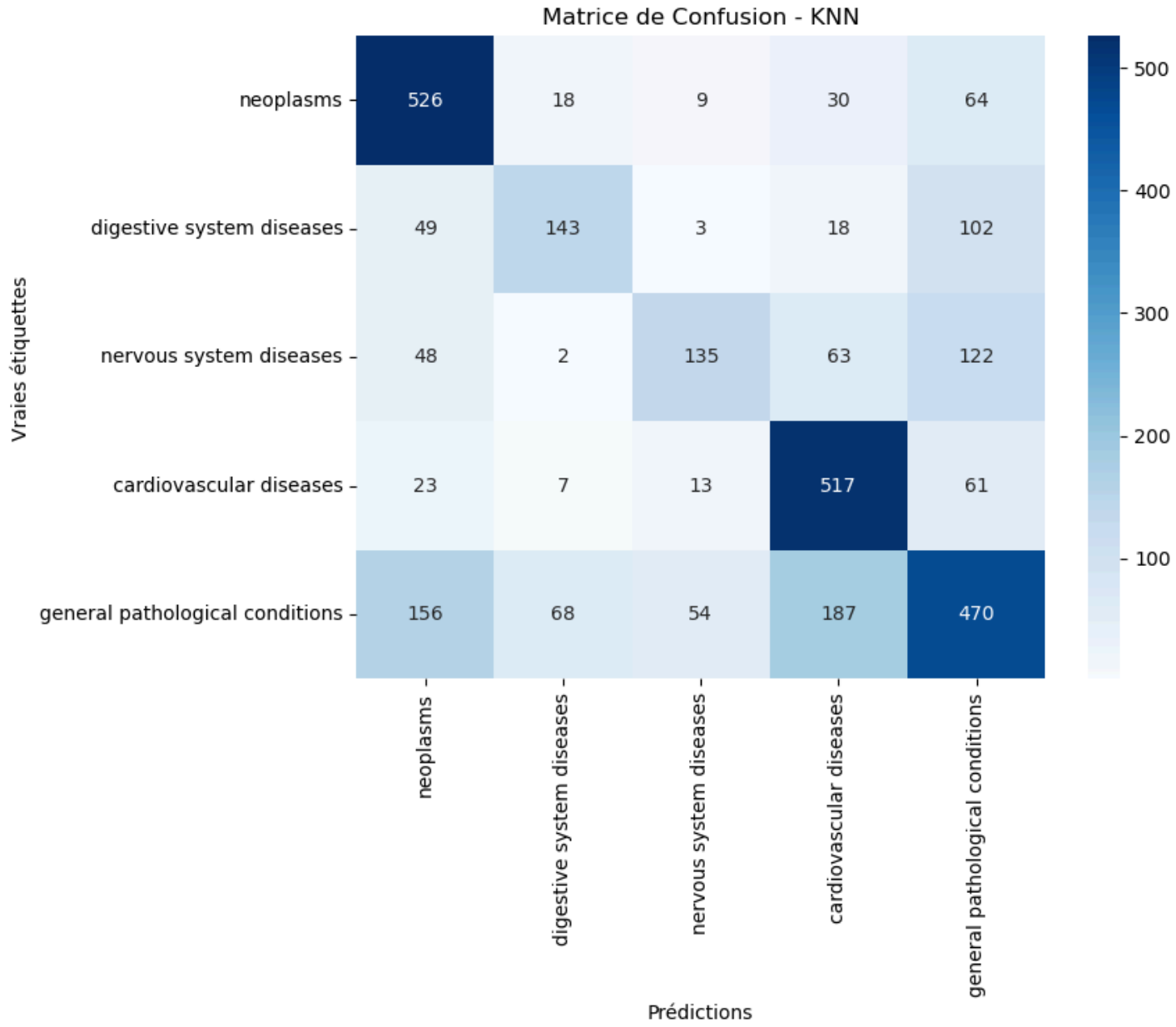
	precision	recall	f1-score	support
neoplasms	0.66	0.81	0.73	647
digestive system diseases	0.60	0.45	0.52	315
nervous system diseases	0.63	0.36	0.46	370
cardiovascular diseases	0.63	0.83	0.72	621
general pathological conditions	0.57	0.50	0.54	935
accuracy			0.62	2888
macro avg	0.62	0.59	0.59	2888
weighted avg	0.62	0.62	0.61	2888

Résultat final :
Avec cet hyperparamètre optimisé, le modèle KNN a atteint une **accuracy de 62.02%**, ce qui représente la meilleure performance obtenue parmi les modèles testés. Ce résultat montre que le modèle KNN, lorsqu'il utilise une large sélection de voisins, est en mesure de bien capturer les relations dans les données et de fournir des prédictions globalement précises.

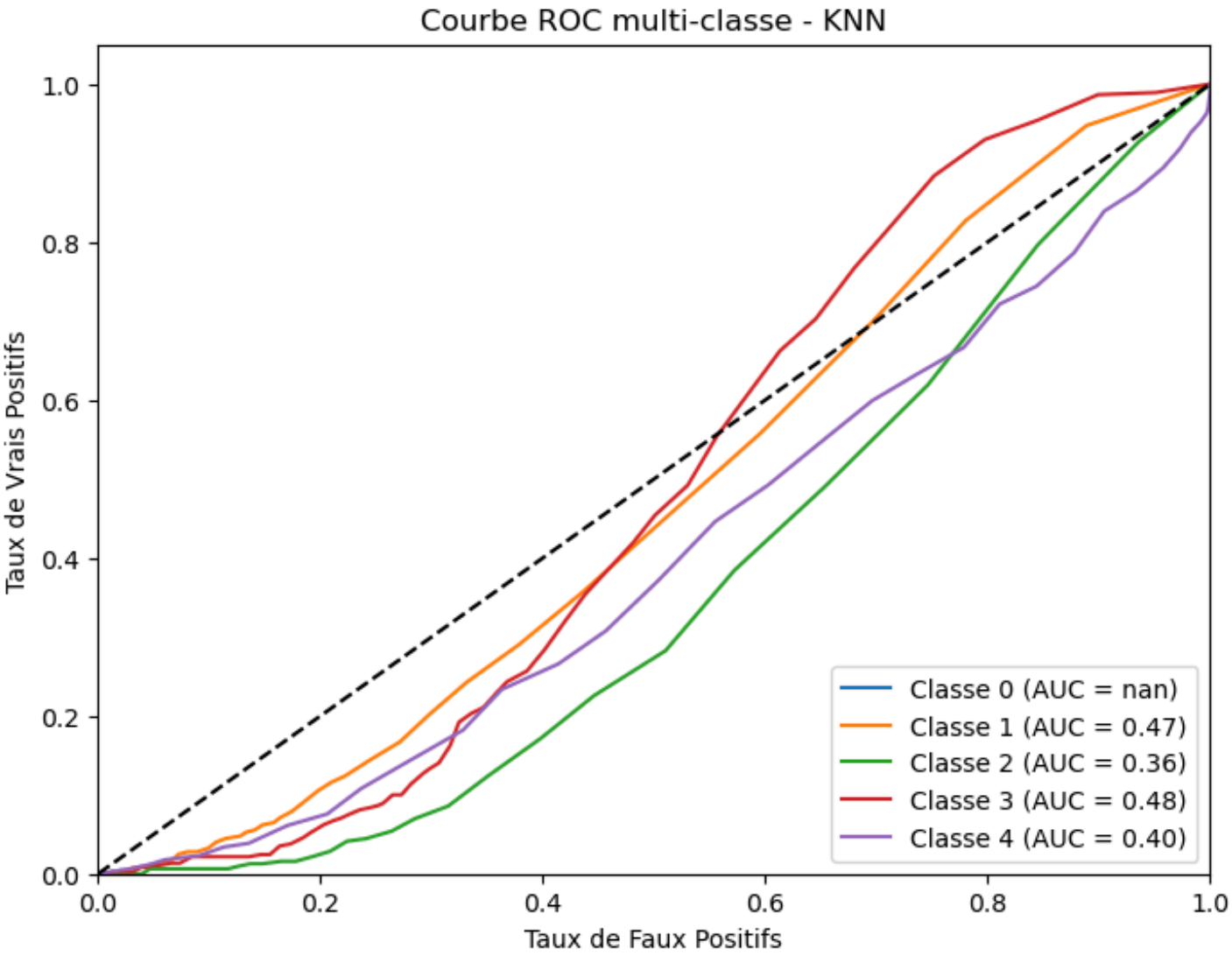
Rapport de classification :
Le modèle fonctionne particulièrement bien pour la classe **neoplasms** avec une précision de 0.66 et un rappel de 0.81, ainsi que pour la classe **cardiovascular diseases** qui atteint une précision de 0.63 et un rappel de 0.83. Ces performances élevées suggèrent que le modèle KNN est capable de capturer des caractéristiques significatives dans ces classes. En revanche, les classes telles que **digestive system diseases** et **nervous system diseases** montrent des

scores plus faibles en rappel, indiquant que le modèle a plus de difficulté à bien classifier ces catégories avec la même configuration de voisins.

Matrice de confusion :



Courbe ROC :


Courbe ROC multi-classe - KNN
Taux de Vrais Positifs
Taux de Faux Positifs
0.0
0.2
0.4
0.6
0.8
1.0
Classe 0 (AUC = nan)
Classe 1 (AUC = 0.47)
Classe 2 (AUC = 0.36)
Classe 3 (AUC = 0.48)
Classe 4 (AUC = 0.40)

3.2.3 K-means

```python
# Modèle 3 : K-means (pour clustering non supervisé)
print("\nModèle K-means (Clustering)")
kmeans_model = KMeans(n_clusters=5, random_state=42)  # Cinq clusters pour les cinq classes
kmeans_model.fit(X_train)

# Utilisation des clusters pour prédire les classes
y_pred_kmeans_train = kmeans_model.predict(X_train)
y_pred_kmeans_test = kmeans_model.predict(X_test)

# Ajustement des labels de K-means pour correspondre à y_train
labels_train = np.zeros_like(y_pred_kmeans_train)
for i in range(5):  # Pour chaque cluster, on trouve la majorité des vraies étiquettes
    mask = (y_pred_kmeans_train == i)
    labels_train[mask] = mode(y_train[mask])[0]

# Appliquer ce mappage aux prédictions du test
labels_test = np.zeros_like(y_pred_kmeans_test)
for i in range(5):
    mask = (y_pred_kmeans_test == i)
    labels_test[mask] = mode(y_train[y_pred_kmeans_train == i])[0]

# Évaluation du modèle K-means
accuracy_kmeans = accuracy_score(y_test, labels_test)
print(f"Accuracy: {accuracy_kmeans * 100:.2f}%")
print("\nClassification Report (K-means):")
print(classification_report(y_test, labels_test, target_names=class_names))
```

Pour le modèle **K-means**, **le nombre de clusters, k, a été fixé à 5**, correspondant aux cinq classes cibles de notre jeu de données. Ce choix permet au modèle de diviser les observations en cinq groupes distincts dans l'espace de caractéristiques, et chaque cluster représente approximativement l'une des classes définies.

Résultats du modèle :

```
Modèle K-means (Clustering)
Accuracy: 42.42%

Classification Report (K-means):
                              precision    recall  f1-score   support

                    neoplasms       0.42      0.62      0.50       647
      digestive system diseases       0.00      0.00      0.00       315
        nervous system diseases       0.00      0.00      0.00       370
        cardiovascular diseases       0.51      0.60      0.55       621
general pathological conditions       0.37      0.49      0.42       935

                     accuracy                           0.42      2888
                    macro avg       0.26      0.34      0.30      2888
                 weighted avg       0.33      0.42      0.37      2888
```

Résultat final :
Avec k = 5, le modèle K-means a obtenu une accuracy de 42.42%. Bien que cette précision soit relativement modeste, elle reflète la capacité limitée du clustering non supervisé à détecter des frontières de classe nettes dans des données nécessitant une approche supervisée.

Rapport de classification :
Le modèle a montré des performances variables selon les classes. Il a bien fonctionné pour les classes **neoplasms** et **cardiovascular diseases**, atteignant des scores de rappel de 0.62 et 0.60 respectivement, ce qui signifie que le modèle a identifié une proportion raisonnable d'échantillons dans ces catégories. Cependant, les autres classes, notamment **digestive system diseases** et **nervous system diseases**, montrent des scores nuls en précision et en rappel, ce qui révèle que le modèle n'a pas réussi à isoler des clusters spécifiques pour ces catégories.

3.2.4 Arbre de décision

```python
# Modèle 4 : Arbre de Décision
print("\nModèle Arbre de Décision")
dt_model = DecisionTreeClassifier(max_depth=32, random_state=42)  # Ajuster les paramètres si nécessaire
dt_model.fit(X_train, y_train)
y_pred_dt = dt_model.predict(X_test)

# Évaluation du modèle Arbre de Décision
accuracy_dt = accuracy_score(y_test, y_pred_dt)
print(f"Accuracy: {accuracy_dt * 100:.2f}%")
print("\nClassification Report (Arbre de Décision):")
print(classification_report(y_test, y_pred_dt, target_names=class_names))
```

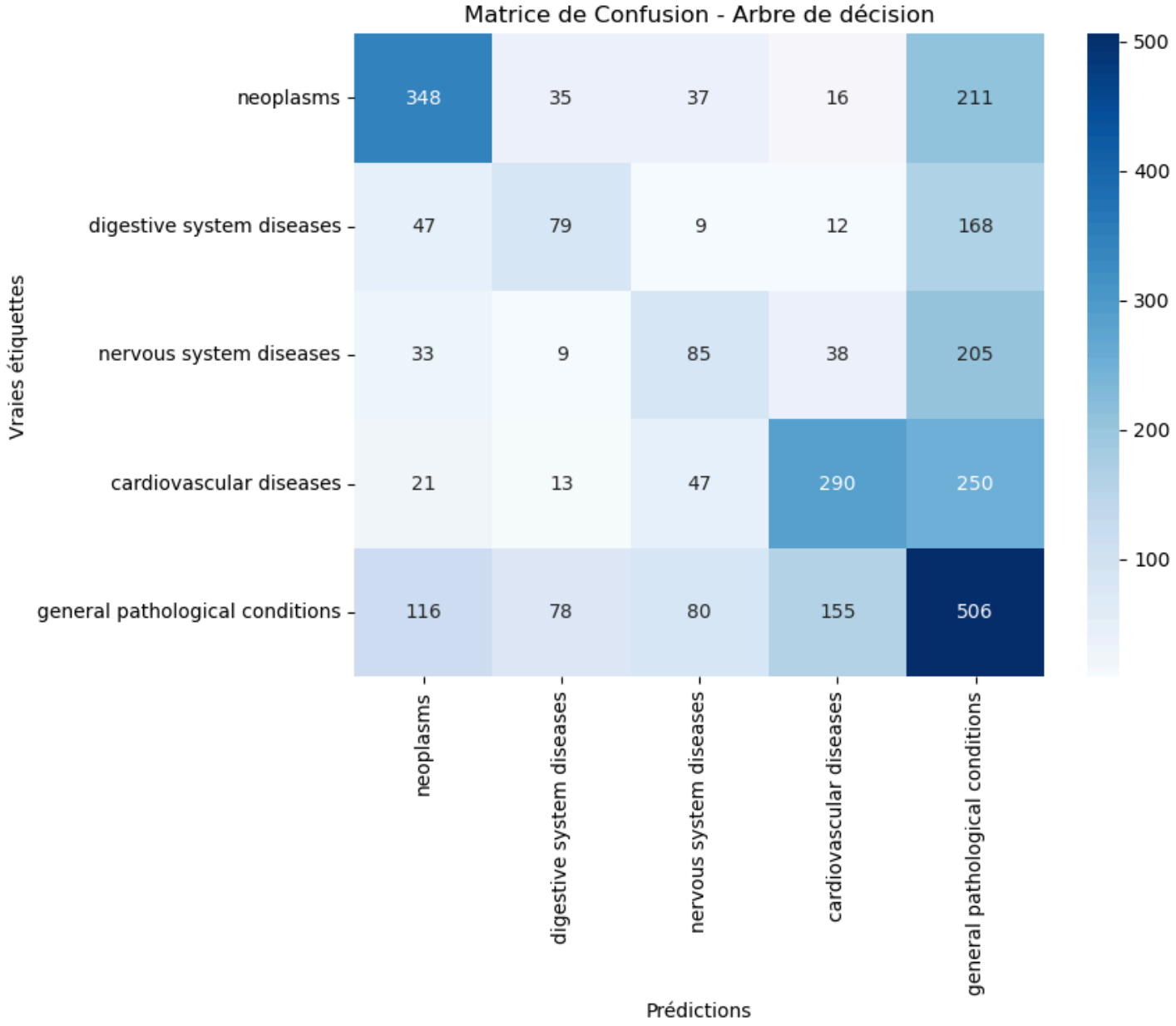
Résultats du modèle :

```
Accuracy: 45.29%

Classification Report (Arbre de Décision):
                              precision    recall  f1-score   support

                    neoplasms       0.62      0.54      0.57       647
      digestive system diseases       0.37      0.25      0.30       315
        nervous system diseases       0.33      0.23      0.27       370
        cardiovascular diseases       0.57      0.47      0.51       621
general pathological conditions       0.38      0.54      0.44       935

                     accuracy                           0.45      2888
                    macro avg       0.45      0.41      0.42      2888
                 weighted avg       0.46      0.45      0.45      2888
```

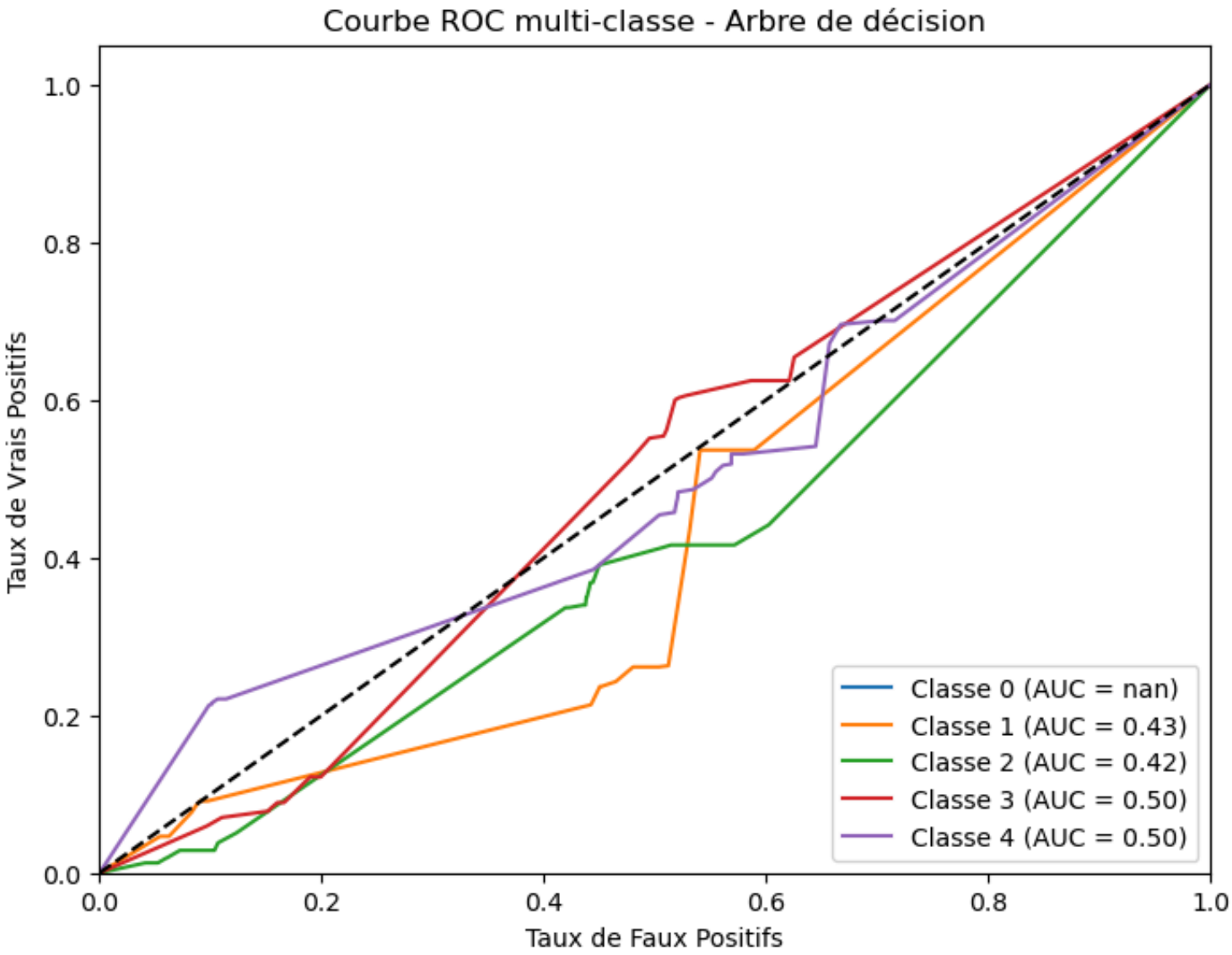
Résultat final :
Le modèle d'arbre de décision a atteint une **accuracy de 45.29%**, ce qui démontre une performance moyenne pour la tâche de classification en raison de la nature des données et de la complexité des relations entre les classes.

Rapport de classification :
Le modèle a bien fonctionné pour les classes **neoplasms** et **cardiovascular diseases**, avec des scores de précision respectifs de 0.62 et 0.57 et des rappels de 0.54 et 0.47, ce qui reflète une capacité raisonnable à identifier des échantillons de ces catégories. Cependant, la performance pour les classes **digestive system diseases** et **nervous system diseases** reste limitée, avec des scores plus faibles en précision et en rappel, ce qui indique que le modèle a des difficultés à isoler des caractéristiques spécifiques pour ces classes.

Matrice de confusion :



Courbe ROC :



3.3 Ajout du modèle BioDeBERTa

Présentation du modèle :

Le modèle **BioDeBERTa (Bio-Domain Enhanced Representation from Transformers with Adapters)** est une version optimisée du modèle **BERT**, spécialisée pour le domaine biomédical. Construit pour gérer des données textuelles biomédicales, **BioDeBERTa** est particulièrement bien adapté pour capturer les nuances et la terminologie spécifique des textes médicaux. Contrairement aux modèles d'apprentissage automatique traditionnels, **BioDeBERTa** utilise une architecture de transformateurs qui permet de mieux comprendre les contextes complexes et les relations entre les termes biomédicaux.

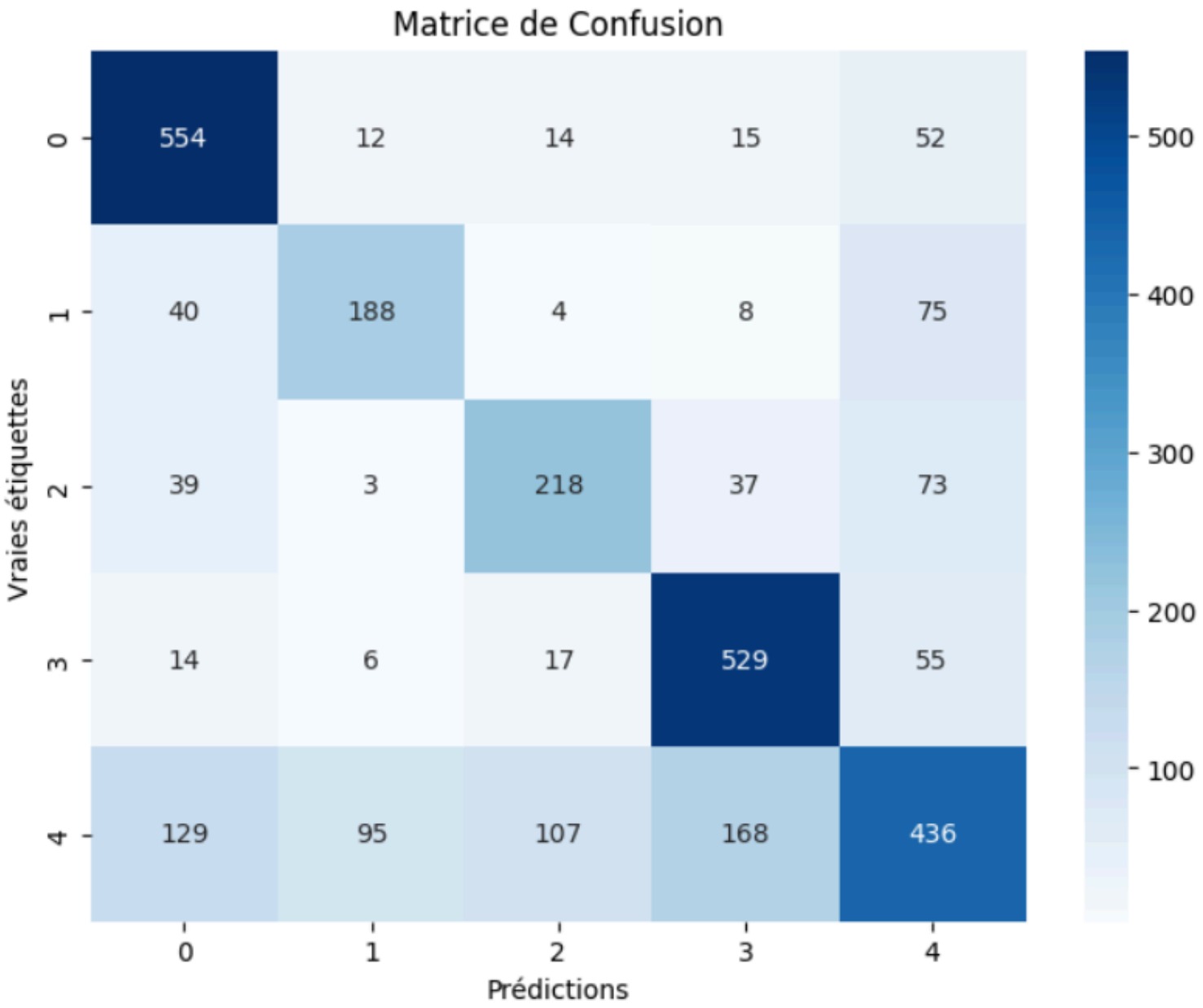
Exploitation dans ce projet :
Dans le cadre de notre projet de classification de textes biomédicaux,

BioDeBERTa a été employé pour tirer parti de sa capacité à comprendre les textes médicaux. En entraînant le modèle sur nos données, nous avons pu bénéficier de ses capacités d'adaptation aux textes biomédicaux, permettant une classification plus fine et pertinente par rapport à nos classes spécifiques.

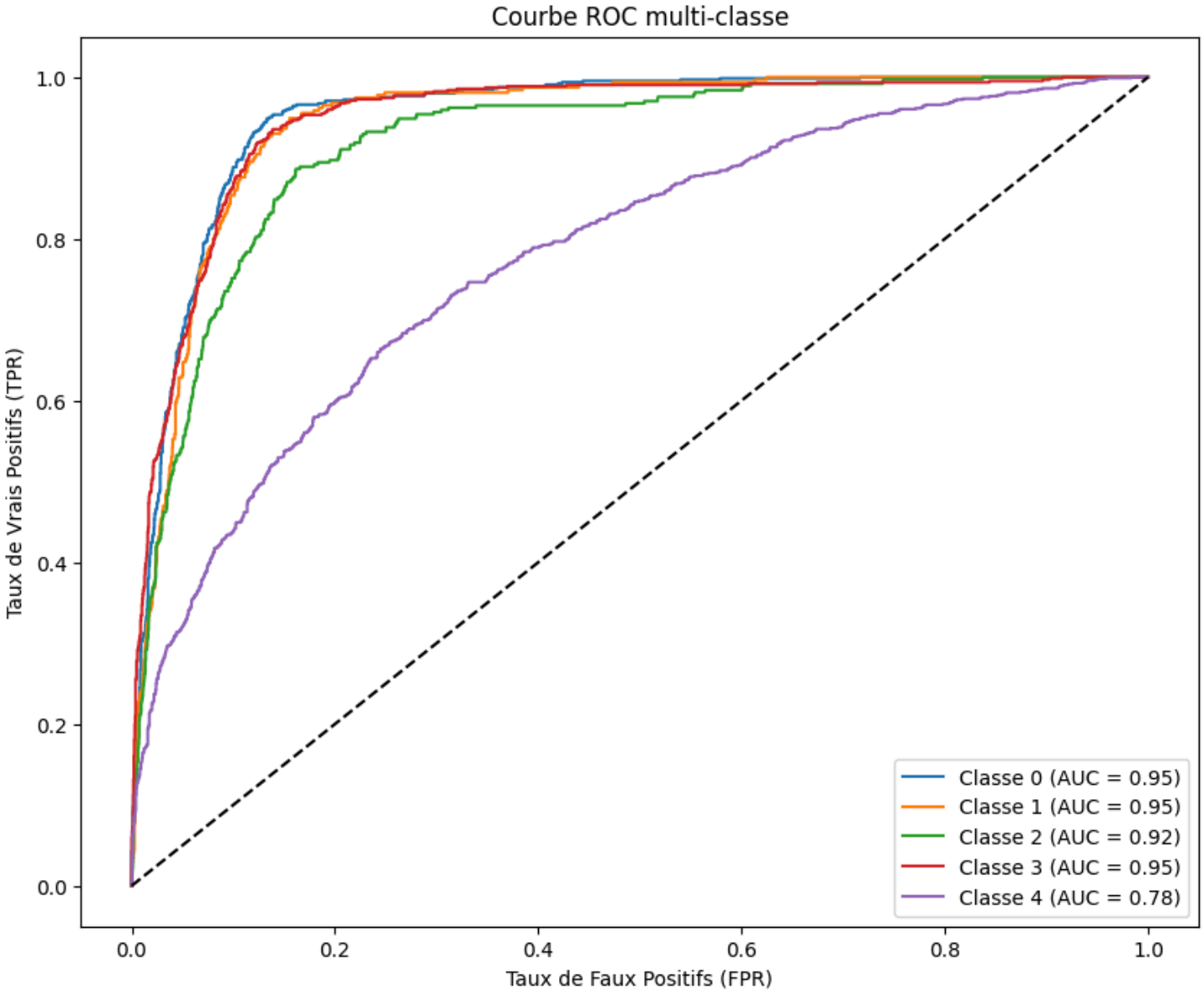
Résultats :
Le modèle **BioDeBERTa** a atteint une **accuracy** de 67% sur nos données, ce qui constitue une amélioration significative par rapport aux autres modèles, notamment les modèles traditionnels comme **l'arbre de décision** ou **K-Nearest Neighbors (KNN)**. L'entraînement s'est déroulé sur une seule époque, aboutissant à une **perte d'entraînement de 0.92** et une **perte de validation de 0.78**. Ces résultats confirment l'efficacité de **BioDeBERTa** pour le traitement de textes biomédicaux, démontrant qu'un modèle pré-entraîné dans le domaine biomédical peut grandement améliorer la performance sur des tâches de classification de texte spécialisé.

Matrice de confusion :

```
Accuracy: 0.67
```



Courbe ROC :



IV - Conclusion

Ce projet a exploré la capacité des technologies de traitement du langage naturel (NLP) et des modèles de machine learning à classifier des documents médicaux en fonction de diverses catégories de maladies. Partant d'un ensemble de données riche en résumés cliniques, nous avons appliqué plusieurs étapes de prétraitement pour normaliser les textes, extrait des caractéristiques textuelles significatives via la vectorisation TF-IDF, puis testé différents modèles de classification. Chaque modèle, qu'il soit supervisé ou non supervisé, a offert une perspective unique sur la prédiction des maladies à partir de documents médicaux.

Les modèles traditionnels comme Random Forest, K-Nearest Neighbors (KNN), et les arbres de décision ont montré des résultats intéressants, bien que limités

en précision pour certaines classes de maladies complexes. Le modèle K-Means, quant à lui, a montré les limites des techniques de clustering non supervisé face aux besoins de classification fine dans le domaine médical.

L'introduction de BioDeBERTa, un modèle de NLP spécialisé dans les textes biomédicaux, a cependant marqué une avancée significative. Avec une précision de 67 %, BioDeBERTa a surpassé les autres modèles, démontrant l'importance d'utiliser des modèles de NLP adaptés au contexte biomédical pour capturer les nuances des textes médicaux. En raison de sa capacité à comprendre la terminologie spécifique et les relations complexes des termes médicaux, BioDeBERTa a permis de classifier plus efficacement les textes, validant l'utilité des modèles pré-entraînés spécialisés dans ce domaine.

En conclusion, ce projet souligne le potentiel des modèles de NLP et de machine learning dans l'amélioration des outils de diagnostic automatisé. Bien que les résultats obtenus soient prometteurs, des améliorations sont envisageables, notamment via un ajustement plus fin des hyperparamètres et l'exploration d'autres architectures de modèles spécialisées pour le domaine médical. Cette étude ouvre ainsi la voie à des applications concrètes pour les systèmes de santé, contribuant à la construction d'outils prédictifs précis et adaptés aux besoins des professionnels de la santé.