Université Hassan II
Faculté des Sciences Ben M'Sick

PROJET DU FIN MODULE

Classement automatique du rapport médical

Présenté par

RAMI Rajae

KARROUM Maroua

MARTAJ Badr

Encadre par :

- Mr.KAICH Oussama
- Mr. BEN LAHMAR EL Habib

Année Universitaire 2024 – 2025

Table de Matières

Table des matières

Résumé

La classification automatique de rapports médicaux représente un enjeu crucial pour l'organisation des données cliniques dans les établissements de santé. Face au volume croissant de documents numériques, l'intégration de solutions basées sur le Deep Learning permet de répondre efficacement aux besoins de tri, d'indexation et de recherche d'information. Ce projet vise à développer un modèle prédictif capable d'identifier automatiquement la spécialité médicale d'un rapport à partir de son contenu textuel.

En s'appuyant sur un jeu de données réel composé de transcriptions médicales anonymisées, plusieurs étapes ont été mises en œuvre : prétraitement linguistique des textes, vectorisation avancée, entraînement de modèles de classification, et évaluation des performances. Le modèle final, basé sur des techniques de traitement du langage naturel (NLP) et des architectures de Deep Learning, offre une précision satisfaisante pour la classification multi-classes des rapports médicaux.

Cette solution constitue un outil d'aide à la gestion documentaire médicale, contribuant à optimiser les processus administratifs et à améliorer l'accessibilité de l'information au sein des systèmes hospitaliers.

Introduction

L’analyse automatique de documents médicaux numériques représente aujourd’hui un enjeu majeur pour les systèmes de santé modernes. La numérisation croissante des rapports cliniques, résultats d'examen ou comptes rendus de consultation impose des solutions permettant un tri et une organisation efficaces de ces documents textuels. Pour les professionnels de santé comme pour les systèmes hospitaliers, disposer d’outils capables de classifier automatiquement ces rapports par spécialité médicale permettrait de fluidifier l’accès à l’information, de gagner du temps et d’améliorer la prise de décision clinique.

Ce projet s’inscrit dans cette logique en développant un modèle prédictif entraîné sur des données réelles issues de transcriptions médicales anonymisées. L’objectif principal est d'automatiser la classification des rapports médicaux en fonction de leur spécialité (comme la cardiologie, la radiologie ou encore la pathologie) afin d’améliorer l’accessibilité, l’organisation et la recherche d’information dans les systèmes de santé.

Le processus de développement s'est articulé autour de plusieurs étapes clés :

Exploration des données : identification des distributions de classes et nettoyage initial.

Prétraitement du texte : nettoyage linguistique, suppression des mots vides.

Vectorisation : transformation des textes en vecteurs numériques via TF-IDF.

Modélisation : entraînement de plusieurs modèles.

Évaluation : comparaison des performances à l’aide de métriques standards (accuracy, F1-score, matrice de confusion).

Le modèle final sélectionné permet de prédire automatiquement la spécialité médicale d’un rapport avec un bon niveau de performance. Il apporte ainsi un soutien concret à l’organisation et au traitement des documents cliniques, en réduisant le temps de classification manuelle et en facilitant l'intégration dans des systèmes hospitaliers numériques. Des perspectives d’amélioration incluent l’intégration de modèles plus avancés, une gestion plus fine du déséquilibre entre classes et la création d’une interface utilisateur via des outils.

CHAPITRE 1

Contexte générale du projet

Ce chapitre comporte une présentation générale du contexte de notre projet de fin de module

1. Contexte générale du projet :

1.1 Problématique :

La problématique principale de ce projet est la suivante :
"Comment classifier automatiquement des rapports médicaux en fonction de leur spécialité à partir de leur contenu textuel, en utilisant des techniques de Deep Learning ?"

Cette question soulève plusieurs sous-problématiques essentielles :

- Comment collecter, nettoyer et préparer efficacement des données textuelles issues de transcriptions médicales ?
- Quels modèles de traitement du langage naturel (NLP) basés sur le Deep Learning sont les plus performants pour ce type de tâche de classification ?
- Quels indicateurs de performance permettent d'évaluer la précision et la robustesse du modèle final dans un contexte multi-classes et potentiellement déséquilibré ?

1.2 Solution :

Ce projet a pour objectif de développer un système de classification automatique capable d'assigner une spécialité médicale (comme la cardiologie, la radiologie, etc.) à un rapport clinique à partir de son contenu textuel.

L'approche s'appuie sur une méthodologie complète, incluant la préparation des données, l'application de techniques avancées de Deep Learning, et l'évaluation comparative de plusieurs modèles pour identifier la solution la plus performante

- L'identification et le nettoyage des transcriptions médicales anonymisées pour un traitement efficace.
- L'application de techniques de NLP modernes (lemmatisation, vectorisation, embeddings) pour extraire des caractéristiques pertinentes du texte.
- L'entraînement et la comparaison de modèles de classification (comme les réseaux de neurones).

L'analyse des performances via des métriques telles que l'accuracy, le F1-score, et les matrices de confusion, afin de sélectionner le modèle le plus fiable.Ce projet vise à fournir un outil concret d'aide à l'organisation documentaire médicale, avec des perspectives d'intégration dans des systèmes hospitaliers intelligents

CHAPITRE 2

Méthodologie et résultats

Ce chapitre résume les étapes suivies dans notre projet de fin de module ainsi que les principaux résultats obtenus.

1. Collecte et Préparation des Données :

1.1 Introduction :

La première étape du projet a consisté à collecter et structurer les données textuelles nécessaires à la modélisation**.** Pour cela, un jeu de données fiable issu de la plateforme Kaggle a été sélectionné, contenant des transcriptions médicales anonymisées, chacune associée à une spécialité médicale. Une fois les données extraites, un nettoyage rigoureux a été réalisé afin d'éliminer les doublons, corriger les incohérences et supprimer les valeurs manquantes. Ensuite, un prétraitement linguistique avancé a été appliqué pour préparer les textes à l'entraînement de modèles de deep learning, incluant la tokenisation, la normalisation du texte et l'encodage des séquences. Ces étapes sont cruciales pour garantir la qualité des entrées textuelles et optimiser la performance des modèles neuronaux utilisés pour la classification.

1.2 Sources et description de données :

Les données utilisées dans ce projet proviennent de la plateforme Kaggle, plus précisément du jeu de données intitulé **"**Medical Transcriptions**"**. Ce jeu comprend de 4999 rapports médicaux anonymisés, chacun étant associé à une spécialité médicale telle que la cardiologie, la radiologie, la neurologie, ou encore la gastroentérologie.

L'extraction des données a été réalisée par téléchargement direct du fichier mtsamples.csv, mis à disposition par le fournisseur sur Kaggle. Ce fichier contient plusieurs colonnes importantes, dont :

- **index** : identifiant unique de chaque échantillon (utilisé comme index dans le DataFrame),
- **description** : courte description ou intitulé du rapport,
- **medical_specialty** : spécialité médicale correspondant au rapport *(*c'est la variable cible du modèle),
- **sample_name** : identifiant ou nom d'origine du fichier de transcription,
- **transcription** : contenu textuel intégral du rapport médical (c'est la variable d'entrée principale),
- **keywords** : mots-clés associés au rapport, parfois bruités ou incomplets.

```
df.shape
```

```
(4999, 6)
```

	Unnamed: 0	description	medical_specialty	sample_name	transcription	keywords
1	0	['23', 'year', 'old', 'white', 'female', 'presents', 'co...	allergy, immunolo...	['allergic', 'rhinitis']	['subjective', '23', 'year', 'old', 'white', '...	['allergy', 'immunology', 'allergic', 'rhinitis', 'allergies', 'asthma', 'nasal', '
2	1	['consult', 'laparoscopic', 'gastric', 'bypass']	bariatrics	['laparoscopic', 'gastric', 'bypass', 'consult...	['past', 'medical', 'history', 'difficulty', '...	['bariatrics', 'laparoscopic', 'gastric', 'bypass', 'weight', 'loss', 'programs',
3	2	['consult', 'laparoscopic', 'gastric', 'bypass']	bariatrics	['laparoscopic', 'gastric', 'bypass', 'consult...	['history', 'present', 'illness', 'seen', 'ab...	['bariatrics', 'laparoscopic', 'gastric', 'bypass', 'heart', 'attacks', 'body', 'w
4	3	['2', 'mode', 'doppler']	cardiovascular, pul...	['2', 'echocardiogram', '1']	['2', 'mode', '1', 'left', 'atrial', 'enlarge...	['cardiovascular', 'pulmonary', '2', 'mode', 'doppler', 'aortic', 'valve', 'atri
5	4	['2', 'echocardiogram']	cardiovascular, pul...	['2', 'echocardiogram', '2']	['1', 'left', 'ventricular', 'cavity', 'size', '...	['cardiovascular', 'pulmonary', '2', 'doppler', 'echocardiogram', 'annular'
6	5	['morbid', 'obesity', 'laparoscopic', 'antecolic', 'a...	bariatrics	['laparoscopic', 'gastric', 'bypass']	['preoperative', 'diagnosis', 'morbid', '...	['bariatrics', 'gastric', 'bypass', 'eea', 'anastomosis', 'roux', 'en', 'antegast.
7	6	['liposuction', 'supraumbilical', 'abdomen', 'revisi...	bariatrics	['liposuction']	['preoperative', 'diagnoses', '1', 'defor...	['bariatrics', 'breast', 'reconstruction', 'excess', 'lma', 'anesthesia', 'lipody
8	7	['2', 'echocardiogram']	cardiovascular, pul...	['2', 'echocardiogram', '3']	['2', 'echocardiogram', 'multiple', 'vie...	['cardiovascular', 'pulmonary', '2', 'echocardiogram', 'cardiac', 'function'
9	8	['suction', 'assisted', 'lipectomy', 'lipodystrophy',...	bariatrics	['lipectomy', 'abdomen', 'thighs']	['preoperative', 'diagnosis', 'lipodystr...	['bariatrics', 'lipodystrophy', 'abd', 'pads', 'suction', 'assisted', 'lipectom.
10	9	['echocardiogram', 'doppler']	cardiovascular, pul...	['2', 'echocardiogram', '4']	['description', '1', 'normal', 'cardiac', 'c...	['cardiovascular', 'pulmonary', 'ejection', 'fraction', 'lv', 'systolic', 'functic

1.3 Nettoyage des données :

✓ La détection des doublons :

Le but de la détection de doublons est d'éliminer les enregistrements répétés pour améliorer la qualité des données.

```
# Affichage avant suppression
print(f"Nombre de lignes AVANT suppression des doublons : {df.shape[0]}")

# Comptage des doublons dans la colonne
nb_doublons = df.duplicated().sum()
print(f"Nombre de doublons trouvés :  {nb_doublons}")

# Suppression des doublons
df = df.drop_duplicates()

# Affichage après suppression
print(f"Nombre de lignes APRÈS suppression des doublons : {df.shape[0]}")
```

```
Nombre de lignes AVANT suppression des doublons : 4999
Nombre de doublons trouvés :  0
Nombre de lignes APRÈS suppression des doublons : 4999
```

✓ La détection des valeurs manquantes :

Le but de la détection des valeurs manquantes est d'assurer l'intégrité des données en identifiant et traitant les absences.

Pour cela, nous avons utilisé la méthode : isnull().sum() afin d'identifier les colonnes concernées.

```
Nombre de valeurs manquantes par colonne :
Unnamed: 0             0
description            0
medical_specialty      0
sample_name            0
transcription         33
keywords            1068
dtype: int64
```

Colonne transcription :

```
df['transcription'] = df['transcription'].fillna('Not transcribed')
```

Il y avait 33 valeurs manquantes dans la colonne transcription, qui ont été remplacées par la valeur unknow. Après cette opération, il n'y a plus de valeurs manquantes dans cette colonne, garantissant ainsi la complétude des données.

Colonne keywords :

```
# 1. Replace empty strings or strings with only spaces by NaN
df["keywords"] = df["keywords"].replace(r'^\s*$', np.nan, regex=True)

# 2. Fill NaN values by the mode of 'keywords' per 'medical_specialty' or 'unknown'
keyword_mode_by_specialty = df.groupby("medical_specialty")["keywords"].transform(
    lambda x: x.fillna(x.mode()[0] if not x.mode().empty else "unknown")
)

# 3. Final replacement in the original column
df["keywords"] = df["keywords"].fillna(keyword_mode_by_specialty)
```

```
print(f"Nombre de 'unknown' dans keywords : {(df['keywords'] == 'unknown').sum()}")
```

```
Nombre de 'unknown' dans keywords : 8
```

Ce code remplace les valeurs vides de la colonne keywords par le mode (valeur la plus fréquente) selon chaque medical_specialty, ou "unknown" si aucun mode n'est disponible.

Il affiche ensuite le nombre de fois où "unknown" apparaît dans la colonne.

Le résultat après nettoyage :

```
Nombre de valeurs manquantes par colonne :
Unnamed: 0          0
description         0
medical_specialty   0
sample_name         0
transcription       0
keywords            0
dtype: int64
```

✓ **Suppression des Caractères Non Alphabétiques et Uniformisation des Casse**

```
# Nettoyage amélioré
for col in df.select_dtypes(include='object').columns:
    df[col] = df[col].apply(lambda text:
        re.sub(r'\s+', ' ',
        re.sub(r'[^1-9a-zA-Z]', ' ', text))     # <= remplace par espace au lieu de supprimer
        .strip().lower() if isinstance(text, str) else text
    )
```

Ce code parcourt toutes les colonnes de type "objet" dans un DataFrame et remplace les caractères non alphabétiques par des espaces, puis met le texte en minuscules. Il nettoie ainsi les données textuelles sans supprimer les valeurs.

✓ **Tokenisation des colonnes de type texte :**

```python
for col in df.select_dtypes(include='object').columns:
    df[col] = df[col].apply(lambda text: text.split() if isinstance(text, str) else text)
    print(f"Exemple de tokens dans la colonne '{col}':")
    print(df[col].iloc[0])  # Affiche la première ligne de la colonne transformée
    print('---')
```

```
Exemple de tokens dans la colonne 'description':
['a', '23', 'year', 'old', 'white', 'female', 'presents', 'with', 'complaint', 'of', 'allergies']
---
Exemple de tokens dans la colonne 'medical_specialty':
['allergy', 'immunology']
---
Exemple de tokens dans la colonne 'sample_name':
['allergic', 'rhinitis']
---
Exemple de tokens dans la colonne 'transcription':
['subjective', 'this', '23', 'year', 'old', 'white', 'female', 'presents', 'with', 'complaint', 'of', 'allergies', 'she', 'used', 'to', 'have', 'allergi
es', 'when', 'she', 'lived', 'in', 'seattle', 'but', 'she', 'thinks', 'they', 'are', 'worse', 'here', 'in', 'the', 'past', 'she', 'has', 'tried', 'clari
tin', 'and', 'zyrtec', 'both', 'worked', 'for', 'short', 'time', 'but', 'then', 'seemed', 'to', 'lose', 'effectiveness', 'she', 'has', 'used', 'allegr
a', 'also', 'she', 'used', 'that', 'last', 'summer', 'and', 'she', 'began', 'using', 'it', 'again', 'two', 'weeks', 'ago', 'it', 'does', 'not', 'appea
r', 'to', 'be', 'working', 'very', 'well', 'she', 'has', 'used', 'over', 'the', 'counter', 'sprays', 'but', 'no', 'prescription', 'nasal', 'sprays', 'sh
e', 'does', 'have', 'asthma', 'but', 'doest', 'not', 'require', 'daily', 'medication', 'for', 'this', 'and', 'does', 'not', 'think', 'it', 'is', 'flarin
g', 'up', 'medications', 'her', 'only', 'medication', 'currently', 'is', 'ortho', 'tri', 'cyclen', 'and', 'the', 'allegra', 'allergies', 'she', 'has',
'no', 'known', 'medicine', 'allergies', 'objective', 'vitals', 'weight', 'was', '13', 'pounds', 'and', 'blood', 'pressure', '124', '78', 'heent', 'her',
'throat', 'was', 'mildly', 'erythematous', 'without', 'exudate', 'nasal', 'mucosa', 'was', 'erythematous', 'and', 'swollen', 'only', 'clear', 'drainag
e', 'was', 'seen', 'tms', 'were', 'clear', 'neck', 'supple', 'without', 'adenopathy', 'lungs', 'clear', 'assessment', 'allergic', 'rhinitis', 'plan',
'1', 'she', 'will', 'try', 'zyrtec', 'instead', 'of', 'allegra', 'again', 'another', 'option', 'will', 'be', 'to', 'use', 'loratadine', 'she', 'does',
'not', 'think', 'she', 'has', 'prescription', 'coverage', 'so', 'that', 'might', 'be', 'cheaper', '2', 'samples', 'of', 'nasonex', 'two', 'sprays', 'i
n', 'each', 'nostril', 'given', 'for', 'three', 'weeks', 'a', 'prescription', 'was', 'written', 'as', 'well']
---
```

Ce code extrait et affiche des tokens (mots ou sous-chaînes) pour chaque colonne de type texte (`object`) dans un DataFrame `df`, en divisant les chaînes de caractères en listes de mots. Il montre un exemple des premiers tokens extraits pour chaque colonne textuelle.

- ✓ **Suppression des stopwords :**

```python
import nltk
from nltk.corpus import stopwords

nltk.download('stopwords')
stop_words = set(stopwords.words('english'))

# Supprimer les stopwords dans chaque colonne texte (chaque cellule contient une liste de mots)
for col in df.select_dtypes(include='object').columns:
    df[col] = df[col].apply(
        lambda tokens: [word for word in tokens if word not in stop_words]
        if isinstance(tokens, list) else tokens
    )
    print(f"Exemple sans stopwords pour la colonne '{col}':")
    print(df[col].iloc[0])  # Affiche un exemple
    print('---')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Hp\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Exemple sans stopwords pour la colonne 'description':
['23', 'year', 'old', 'white', 'female', 'presents', 'complaint', 'allergies']
---
Exemple sans stopwords pour la colonne 'medical_specialty':
['allergy', 'immunology']
---
Exemple sans stopwords pour la colonne 'sample_name':
['allergic', 'rhinitis']
---
```

Télécharge les stopwords en anglais via NLTK et supprime ces mots peu informatifs de chaque colonne textuelle tokenisée.

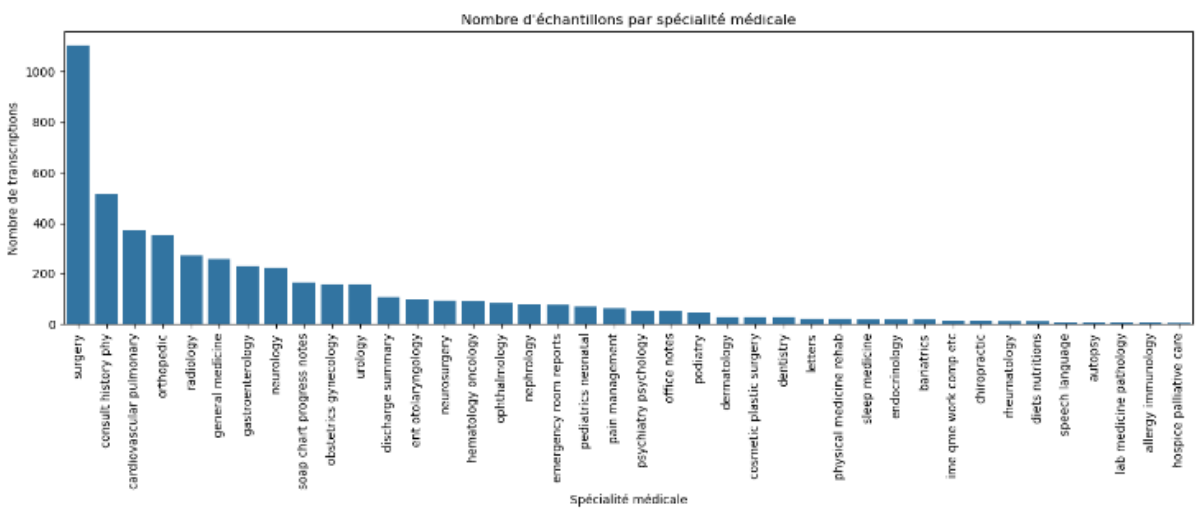
2. Analyse Exploratoire des Données (EDA) :

2.1 Introduction :

L'analyse exploratoire de données, abrégée en AED, est un processus qui rend compte en détail de la structure, de la relation, et de la dynamique d'un jeu de données avant de les mapper avec les modèles prédictifs ou les algorithmes de machine Learning. Il permet de déceler les déviations, les missing values, les distributions des variables, et les corrélations potentielles.

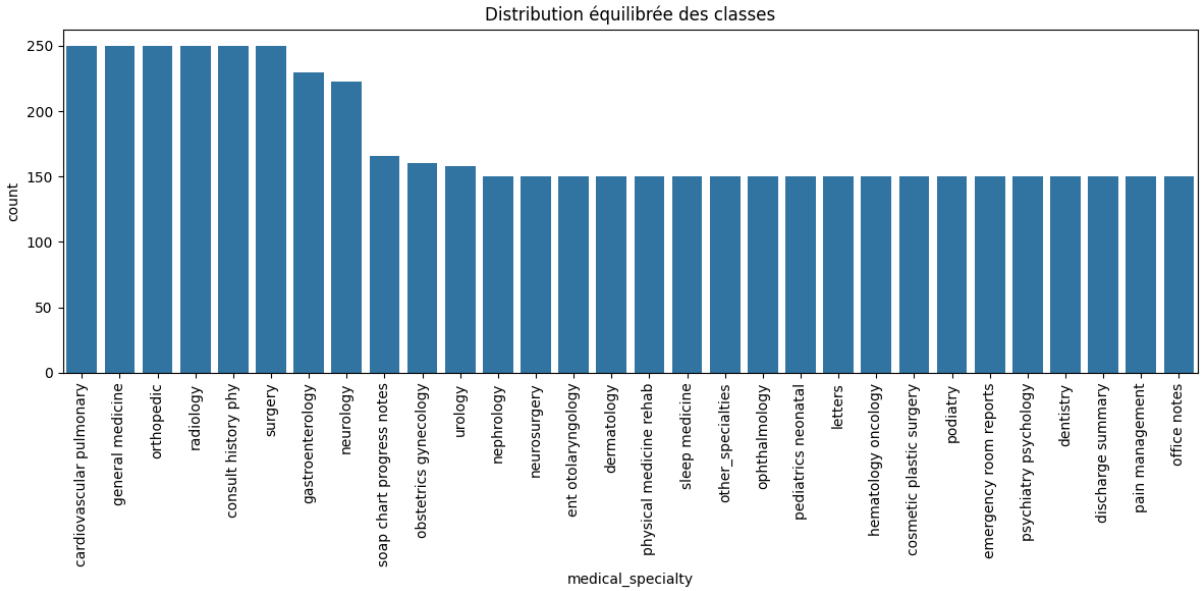
2.2 Démonstration des graphes :

Distribution des rapports médicaux selon la spécialité :



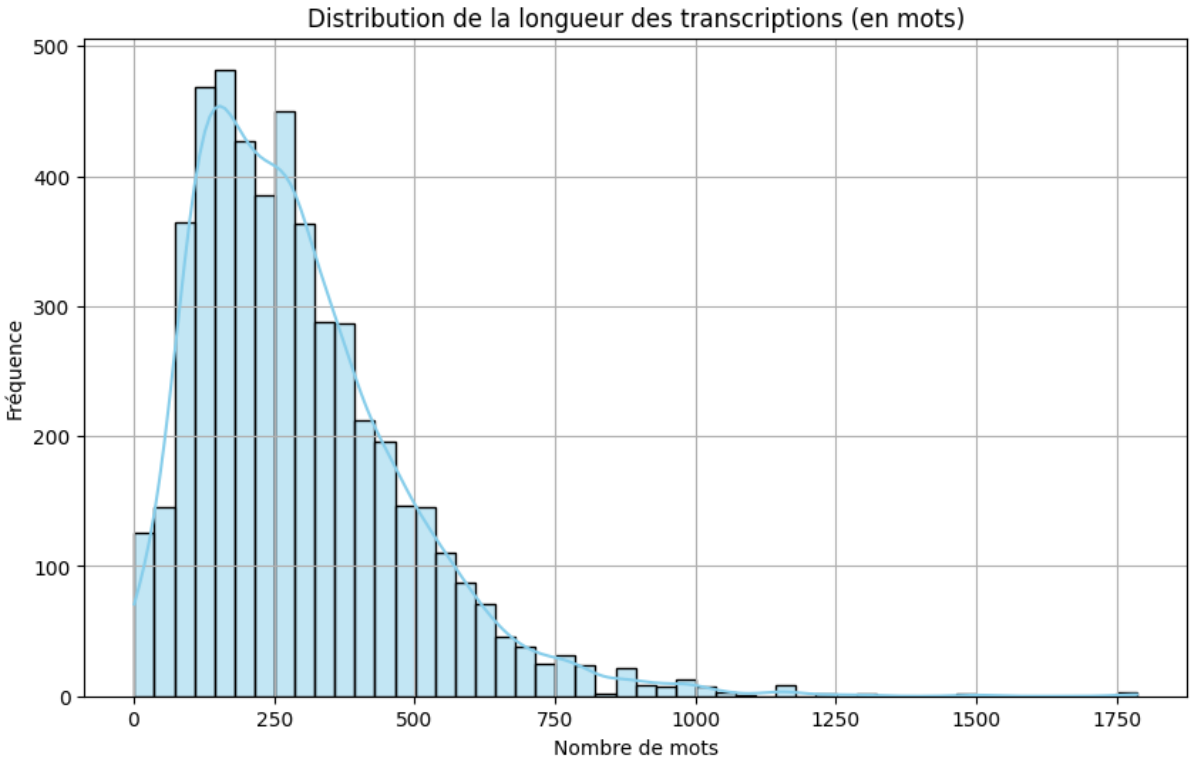
Le graphique montre une **répartition inégale** des transcriptions médicales entre les différentes spécialités. Certaines spécialités comme **Surgery**, **Internal Medicine** ou **Radiology** possèdent un **grand nombre d'échantillons**, ce qui suggère qu'elles sont **plus fréquemment documentées** dans le dataset. À l'inverse, plusieurs spécialités apparaissent avec un **nombre très faible d'occurrences**, voire marginal.

Équilibrage des données :



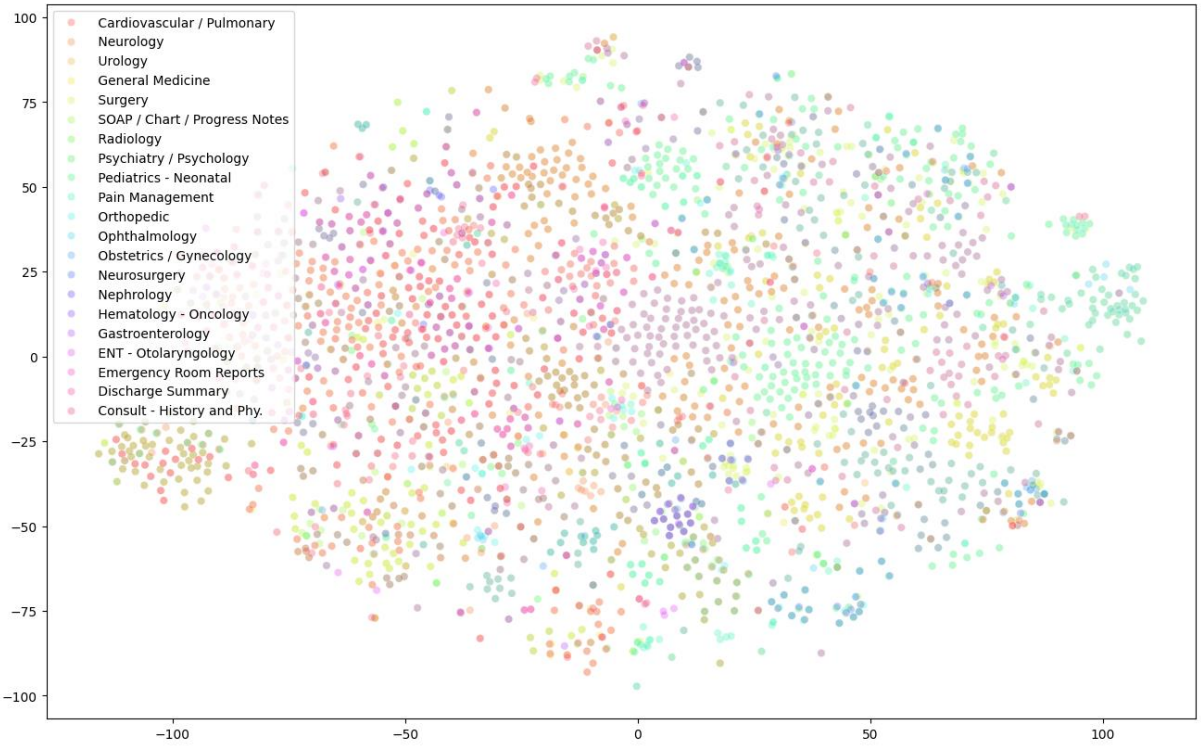
L'objectif est de rééquilibrer les classes de la variable medical_specialty en limitant le nombre d'exemples pour les classes trop fréquentes (à 250) et en augmentant celui des classes rares (à au moins 150), afin d'obtenir un ensemble de données équilibré où chaque spécialité médicale est représentée de manière équitable, ce qui améliore la performance.

Distribution de la longueur des transcriptions (en mots) :



La distribution de la longueur des transcriptions est asymétrique à droite **(**positivement asymétrique)**,** ce qui indique qu'il existe un grand nombre de transcriptions relativement courtes, tandis que certaines peuvent être très longues. La majorité des transcriptions comptent entre 100 et 400 mots, avec un pic de fréquence autour de 200 mots. Au-delà de 400 mots, la fréquence diminue rapidement, et les transcriptions de plus de 1000 mots sont rares. On observe également quelques valeurs extrêmes dépassant 1500 mots, indiquant la présence de transcriptions exceptionnellement longues**.**

Distribution des documents cliniques selon leur contenu textuel :



Certains types de documents ont tendance à **se** regrouper**,** comme les rapports de Radiology**,** Psychiatry ou **S**urgery, ce qui suggère qu'ils partagent des structures linguistiques ou un vocabulaire similaire. Toutefois, on observe également un chevauchement entre plusieurs catégories, probablement dû à des contenus proches ou à l'usage de terminologie médicale commune entre différentes spécialités. Enfin, la dispersion des points au sein des catégories reflète la diversité des textes, même à l'intérieur d'un même type de document.

3. Modélisation, Entraînement & validation:

✓ **Division des Données :**

Avant d'entraîner un modèle en deep Learning, les données sont divisées en deux ensembles : un ensemble d'entraînement (80 %) pour ajuster les paramètres du modèle, et un ensemble de test (20 %) pour évaluer sa capacité à généraliser sur des données nouvelles.

```
# Train/test split
X_train, X_test, y_train, y_test = train_test_split(
    X_res, y_res, test_size=0.2, random_state=42, stratify=y_res
)
```

✓ **Entraînement du modèle :**

```
# Build model
model = Sequential([
    Input(shape=(X_train.shape[1],)),
    Dense(512, activation='relu'),
    Dropout(0.4),
    Dense(256, activation='relu'),
    Dropout(0.3),
    Dense(len(specialties), activation='softmax')
])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Early stopping
early_stop = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True)

# Train model
history = model.fit(
    X_train, y_train,
    epochs=10,
    batch_size=64,
    validation_split=0.2,
    callbacks=[early_stop]
)
```

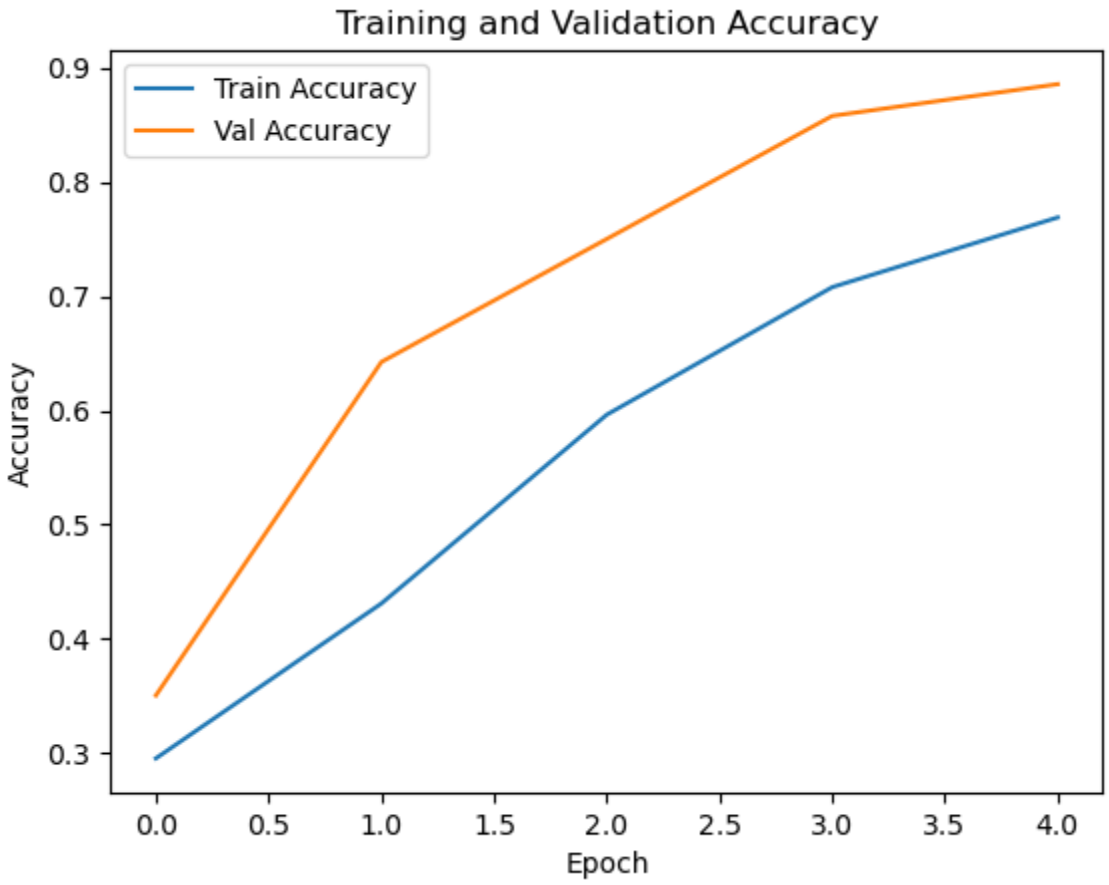
Ce modèle de réseau de neurones permet de classer automatiquement des documents médicaux selon leur spécialité. Il s'appuie sur plusieurs couches avec mécanismes de régularisation pour améliorer la généralisation, et utilise une phase de validation pour ajuster l'entraînement. L'objectif est d'identifier la spécialité médicale à partir du contenu textuel de chaque document.

✓ **Validation :**

```
Epoch 1/5
162/162 ━━━━━━━━━━━━━━━━━━━━ 14s 75ms/step - accuracy: 0.2319 - loss: 3.2851 - val_accuracy: 0.3507 - val_loss: 1.9961
Epoch 2/5
162/162 ━━━━━━━━━━━━━━━━━━━━ 11s 71ms/step - accuracy: 0.3963 - loss: 1.9748 - val_accuracy: 0.6424 - val_loss: 1.5016
Epoch 3/5
162/162 ━━━━━━━━━━━━━━━━━━━━ 11s 69ms/step - accuracy: 0.5485 - loss: 1.5930 - val_accuracy: 0.7500 - val_loss: 0.9517
Epoch 4/5
162/162 ━━━━━━━━━━━━━━━━━━━━ 11s 69ms/step - accuracy: 0.6881 - loss: 1.1539 - val_accuracy: 0.8576 - val_loss: 0.7353
Epoch 5/5
162/162 ━━━━━━━━━━━━━━━━━━━━ 11s 69ms/step - accuracy: 0.7724 - loss: 0.8921 - val_accuracy: 0.8854 - val_loss: 0.5715
23/23 ━━━━━━━━━━━━━━━━━━━━ 1s 35ms/step - accuracy: 0.8826 - loss: 0.5708
Test Accuracy: 87.36%
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 209ms/step
Predicted Specialty: urology
```

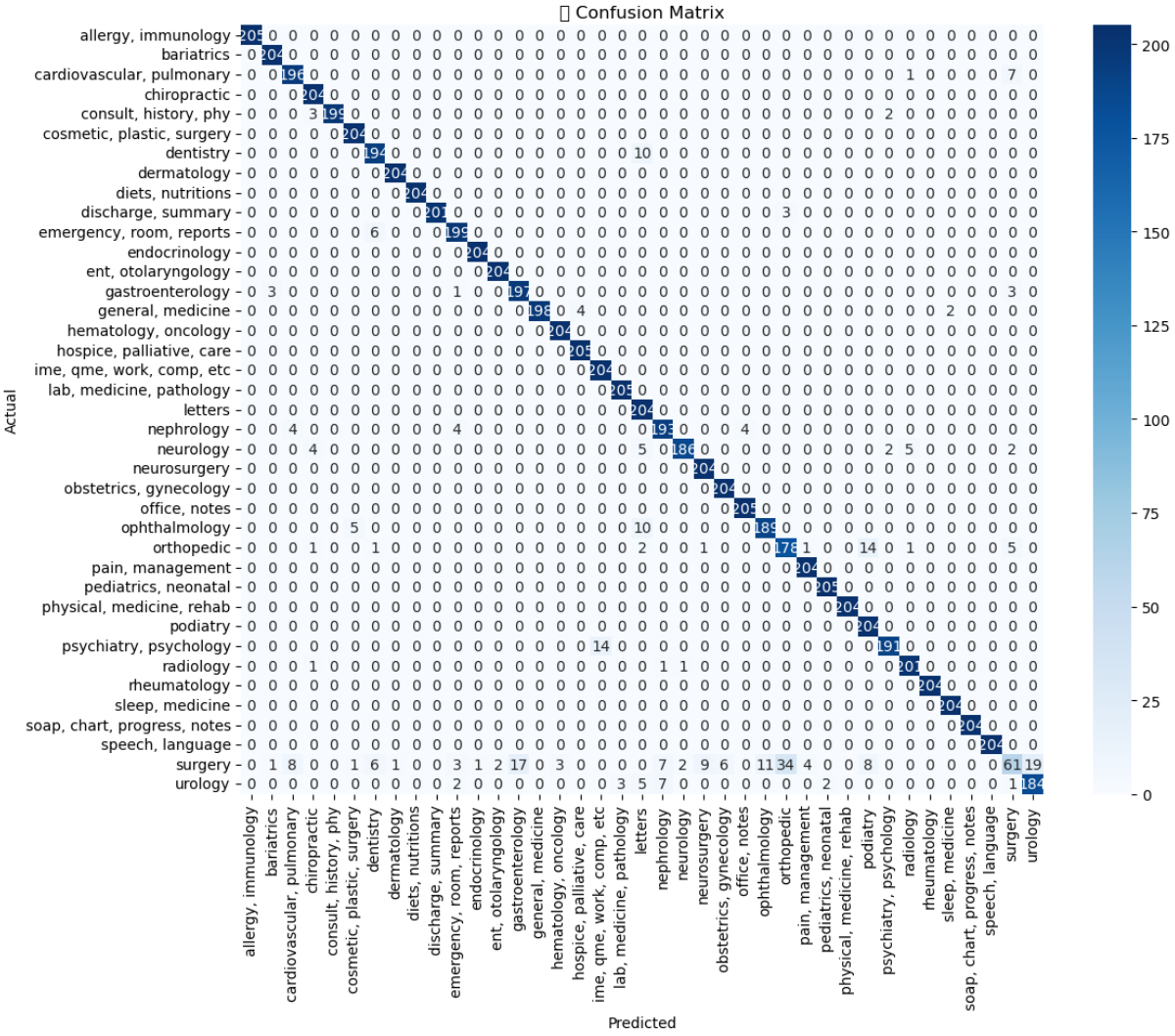
Le modèle de classification a été entraîné sur 5 époques avec une amélioration continue de la précision (jusqu'à 88,54 % en validation) et une diminution des pertes, atteignant une précision de 87,36 % sur les données de test, ce qui montre une bonne généralisation ; il a prédit la spécialité "urology" pour un rapport médical donné.

✓ **Progression de la précision du modèle :**



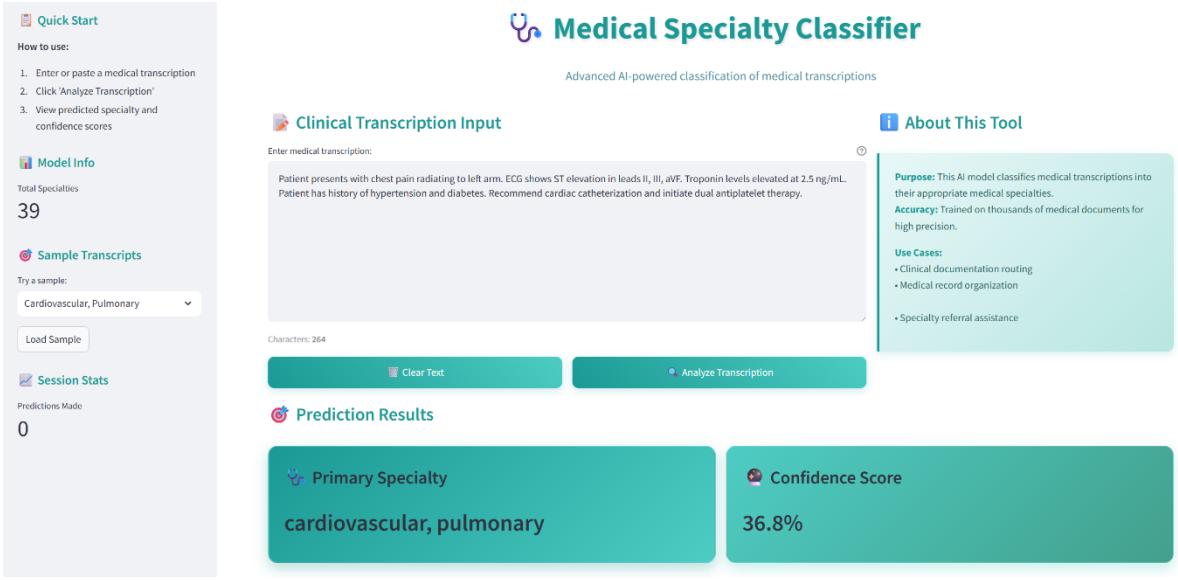
Le graphique montre une amélioration continue de la précision du modèle FCNN, atteignant 87% en validation sans sur apprentissage.

✓ **Matrice de Confusion :**



La matrice de confusion montre une bonne performance globale du modèle avec des prédictions majoritairement correctes.

4. Déploiement & interface :



Cette interface intuitive permet aux utilisateurs (cliniciens, développeurs ou personnel médical) d'entrer des textes cliniques et d'obtenir immédiatement la spécialité médicale correspondante avec un score de confiance. Elle facilite ainsi l'intégration de l'IA dans les processus hospitaliers pour améliorer la gestion et l'analyse documentaire.

Conclusion

Ce projet a permis de concevoir un système fiable de classification automatique des rapports médicaux selon leur spécialité, répondant ainsi à un besoin croissant dans les systèmes de santé numériques. En s'appuyant sur un jeu de données réel et anonymisé, plusieurs étapes ont été rigoureusement mises en œuvre : prétraitement, vectorisation, modélisation et évaluation. Le modèle, sélectionné pour sa performance (notamment en précision et en capacité de généralisation), permet de classifier avec efficacité la spécialité médicale d'un rapport, réduisant considérablement le temps et les efforts requis pour une classification manuelle.

Cette automatisation contribue non seulement à une meilleure organisation des documents cliniques, mais aussi à un accès plus rapide et plus pertinent à l'information pour les professionnels de santé. Elle s'intègre parfaitement dans une logique d'amélioration continue des systèmes hospitaliers, en facilitant la recherche d'information et la prise de décision.

Par ailleurs, le projet ouvre des perspectives intéressantes, notamment l'intégration de modèles plus puissants pour un meilleur traitement du langage naturel, l'optimisation de la gestion du déséquilibre entre les classes, et la mise en place d'une interface interactive pour un déploiement pratique et accessible aux utilisateurs finaux.

Références

[1] “Medical Transcriptions.” Accessed: May 31, 2025. [Online]. Available: https://www.kaggle.com/datasets/tboyle10/medicaltranscriptions