



Royaume du Maroc  
UNIVERSITÉ MOHAMED V - RABAT  
ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE  
ET D'ANALISE DES SYSTÈMES

# Rapport du projet Machine learning, Data Visualisation et Devops

## ”Audio Emotion Analysis”

**Filière :** Génie de la Data (GD)

**Réalisé par :**  
CHAOUA Wiam  
BOUGOUTAYA Hiba ALLAH  
OULHAJ Salwa

**Encadré par :**  
MME S.ELFKIHI  
MME M.ABIK  
MR M.ELHAMLAOUI

**Année Scolaire 2024/2025**



# Remerciements

Nous souhaitons exprimer notre profonde gratitude à Madame Sanaa El Fkihi, notre chef de filière et professeur de machine learning, à Madame Mounia Abik, professeure de datavisualisation, et à Monsieur M. Elhamlaoui, professeur de DevOps, pour leur soutien indéfectible et leur encadrement précieux tout au long de la réalisation de notre projet.

Leur expertise, leur disponibilité et leurs conseils éclairés ont été déterminants pour mener à bien cette initiative. Ils ont su, avec pédagogie et bienveillance, nous guider à chaque étape, en nous transmettant non seulement des connaissances théoriques solides, mais également des compétences pratiques essentielles pour le développement de notre projet.

Nous leur adressons nos sincères remerciements pour la confiance qu'ils nous ont accordée et pour l'inspiration qu'ils ont su insuffler dans notre travail.

# Résumé

Ce projet combine les domaines du Machine Learning, de la Visualisation de Données et du DevOps pour effectuer une analyse des sentiments à partir d'audios. En utilisant un dataset, contenant des conversations annotées émotionnellement, nous avons développé un modèle de classification pour détecter des émotions telles que la joie, la tristesse, la colère, la surprise, la peur, le neutre, et le dégoût. Le projet inclut des étapes rigoureuses de préparation des données, d'extraction de caractéristiques et de modélisation. Une visualisation approfondie des données a été réalisée pour interpréter les résultats, tandis que l'intégration DevOps a permis d'automatiser et de conteneuriser l'application pour un déploiement efficace. Les résultats montrent des performances prometteuses avec des perspectives d'amélioration et d'optimisation.

# Abstract

This project combines the fields of Machine Learning, Data Visualization, and DevOps to perform sentiment analysis from audio data. Using a dataset containing emotionally annotated conversations, we developed a classification model to detect emotions such as joy, sadness, anger, surprise, fear, neutrality, and disgust. The project includes rigorous steps for data preparation, feature extraction, and modeling. Comprehensive data visualization was conducted to interpret the results, while DevOps integration enabled the automation and containerization of the application for efficient deployment. The results demonstrate promising performance with opportunities for further improvement and optimization.

# Introduction Générale

Dans un monde où les interactions humaines et technologiques deviennent de plus en plus interconnectées, la capacité d'analyser et de comprendre les émotions et les sentiments exprimés dans la voix humaine est essentielle. Le projet “Audio Sentiment Analysis” s'inscrit dans ce contexte, en explorant les technologies avancées de machine learning et de data visualisation pour déduire des sentiments à partir d'enregistrements audio.

L'objectif principal de ce projet est de développer un modèle qui permet le traitement des données audio, l'extraction de caractéristiques (features), la modélisation avec des algorithmes de machine learning, et une visualisation claire des résultats. En parallèle, une approche DevOps a été adoptée pour assurer la scalabilité et la maintenabilité du projet à travers des conteneurs et des pipelines d'intégration continue.

Ce rapport présente les étapes clés du projet, de la préparation des données à l'évaluation des performances des modèles, en passant par l'intégration dans un environnement conteneurisé. Il met également en lumière les défis rencontrés, les solutions mises en œuvre, et les perspectives offertes par l'analyse des émotions dans les données audio.

# Table de matières

|  |              |
|--|--------------|
| <b>Remerciements</b>   | <b>2</b>     |
| <b>Résumé</b>  | <b>3</b>     |
| <b>Abstract</b>  | <b>4</b>     |
| <b>Introduction Générale</b>                                   | <b>5</b>     |
| <b>Table des figures</b>                                       | <b>8</b>     |
| <br><b>I Machine Learning</b>                                  | <br><b>9</b> |
| <b>1 MELD Dataset</b>  | <b>10</b>    |
| <b>2 CREMA-D x TESS x SAVEE Dataset</b>                        | <b>11</b>    |
| 2.1 Description des données . . . . .                          | 11           |
| 2.1.1 CREMA-D Dataset . . . . .                                | 11           |
| 2.1.2 TESS Dataset . . . . .                                   | 11           |
| 2.1.3 SAVEE Dataset . . . . .                                  | 12           |
| 2.2 Data Preprocessing . . . . .                               | 13           |
| 2.2.1 Prétraitement des données SAVEE . . . . .                | 13           |
| 2.2.2 Prétraitement des données TESS . . . . .                 | 13           |
| 2.2.3 Prétraitement des données CREMA-D . . . . .              | 13           |
| 2.2.4 Fusion des ensembles de données . . . . .                | 13           |
| 2.2.5 Visualisation des données . . . . .                      | 14           |
| 2.2.6 Exportation des données . . . . .                        | 14           |
| 2.3 Feature Extraction . . . . .                               | 15           |
| 2.3.1 Taux d'Échantillonnage . . . . .                         | 15           |
| Les Taux d'Échantillonnage dans notre Jeu de Données . . . . . | 15           |
| 2.3.2 Techniques d'Augmentation des Données . . . . .          | 15           |
| 2.3.3 Techniques d'Extraction des Caractéristiques . . . . .   | 15           |
| 2.3.4 Création des Vecteurs de Caractéristiques . . . . .      | 16           |
| 2.4 Construction des Modèles . . . . .                         | 17           |
| 2.4.1 Validation Croisée . . . . .                             | 17           |
| 2.4.2 Modèle Utilisé : Régression Logistique . . . . .         | 17           |
| 2.4.3 Pipeline de Pré-traitement . . . . .                     | 17           |

|            |  |           |
|------------|--|-----------|
| 2.5        | Évaluation des Modèles . . . . .           | 18        |
| 2.5.1      | Durée de Validation Croisée . . . . .      | 18        |
| 2.5.2      | Optimisation des Hyperparamètres . . . . . | 18        |
| 2.5.3      | Performance des Modèles . . . . .          | 18        |
| <b>II</b>  | <b>Data Visualisation</b>                  | <b>20</b> |
| <b>3</b>   | <b>Data Preprocessing</b>                  | <b>21</b> |
| <b>4</b>   | <b>Audio Features</b>                      | <b>24</b> |
| 4.1        | Waveplot . . . . .                         | 24        |
| 4.2        | Spectrogramme STFT . . . . .               | 25        |
| 4.3        | MFCC . . . . .                             | 25        |
| 4.4        | Spectrogramme de Mel . . . . .             | 26        |
| 4.5        | Zero Crossing Rate (ZCR) . . . . .         | 26        |
| <b>5</b>   | <b>Le Modèle</b>                           | <b>27</b> |
| <b>III</b> | <b>Devops</b>                              | <b>29</b> |
| <b>6</b>   | <b>Méthode Agile</b>                       | <b>30</b> |
| <b>7</b>   | <b>API avec FastAPI</b>                    | <b>31</b> |
| <b>8</b>   | <b>Docker</b>                              | <b>32</b> |
|            | <b>Conclusion générale</b>                 | <b>35</b> |



# Table des figures

|      |   |    |
|------|---|----|
| 1.1  | Performances des modèles de deep learning sur le dataset MELD . . . . . | 10 |
| 2.1  | Distribution des émotions dans le dataset CREMA-D . . . . .             | 11 |
| 2.2  | Exemples d'enregistrements audio du dataset TESS . . . . .              | 12 |
| 2.3  | Exemples d'enregistrements du dataset SAVEE . . . . .                   | 12 |
| 2.4  | Distribution des émotions dans l'ensemble fusionné . . . . .            | 14 |
| 2.5  | Table des features + label. . . . .                                     | 16 |
| 2.6  | Index de chaque type de features. . . . .                               | 16 |
| 2.7  | Distribution de variable. . . . .                                       | 17 |
| 2.8  | Durée totale de validation croisée. . . . .                             | 18 |
| 2.9  | Meilleurs Hyperparamètres et fitting du modèle. . . . .                 | 18 |
| 2.10 | Résultats . . . . .   | 19 |
| 2.11 | Courbe ROC pour l'ensemble d'entraînement avec l'AUC associée. . . . .  | 19 |
| 2.12 | Courbe ROC pour l'ensemble de test avec l'AUC associée. . . . .         | 19 |
| 3.1  | Distribution des émotions dans la base de données . . . . .             | 21 |
| 3.2  | Impact de la standardisation des données . . . . .                      | 22 |
| 3.3  | Répartition des données après PCA . . . . .                             | 23 |
| 4.1  | Waveplot pour une émotion donnée . . . . .                              | 24 |
| 4.2  | Spectrogramme STFT pour une émotion donnée . . . . .                    | 25 |
| 4.3  | Coefficients MFCC pour une émotion donnée . . . . .                     | 25 |
| 4.4  | Spectrogramme de Mel pour une émotion donnée . . . . .                  | 26 |
| 4.5  | Taux de passage à zéro (ZCR) pour une émotion donnée . . . . .          | 26 |
| 5.1  | Courbe ROC pour l'ensemble d'entraînement . . . . .                     | 27 |
| 5.2  | Courbe ROC pour l'ensemble de test . . . . .                            | 28 |
| 6.1  | Méthode agile : Kanban . . . . .  | 30 |
| 7.1  | API avec FastAPI . . . . .  | 31 |
| 8.1  | Fichier Docker . . . . .  | 32 |
| 8.2  | Fichier requirements.txt . . . . .                                      | 33 |
| 8.3  | Construction de l'image Docker . . . . .                                | 33 |
| 8.4  | Image sur Docker . . . . .  | 34 |
| 8.5  | Exécution de l'image Docker . . . . .                                   | 34 |

**Première partie**  
**Machine Learning**

# MELD Dataset

Le MELD (Multimodal EmotionLines Dataset) est un ensemble de données multimodal constitué de dialogues émotionnels provenant de conversations en ligne. Il inclut à la fois des transcriptions textuelles et des informations émotionnelles provenant de plusieurs modalités, telles que l'audio et la vidéo. Le but principal du MELD est de fournir un terrain d'entraînement pour des modèles capables de comprendre et de classer des émotions humaines dans des contextes conversationnels variés.

Au départ, notre intention était de travailler directement avec ce dataset en utilisant des modèles de machine learning (SVM, XGBoost, LR, RF), afin de tirer parti de la richesse des données multimodales. Cependant, après plusieurs essais, nous avons observé que même les performances des modèles de deep learning étaient très faibles, avec des scores inférieurs à 0,7. Cette faiblesse des résultats suggère que même les modèles de deep learning n'ont pas réussi à exploiter efficacement les informations présentes dans ce dataset pour prédire les émotions de manière précise.

## Emotion Recognition in Conversation on MELD

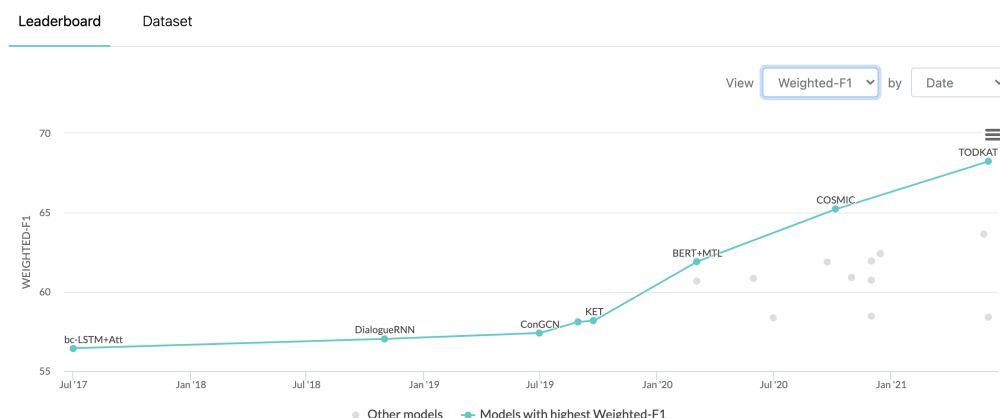


FIGURE 1.1 – Performances des modèles de deep learning sur le dataset MELD

Face à ces résultats décevants, nous avons décidé de réorienter notre approche vers une méthode plus simple et plus accessible, en utilisant un autre dataset. Cette réorientation nous a permis d'obtenir des résultats prometteurs, offrant ainsi un nouvel espoir d'améliorer la classification des émotions dans les dialogues.

# CREMA-D x TESS x SAVEE Dataset

## 2.1 Description des données

Les datasets CREMA-D, TESS et SAVEE sont des ensembles de données utilisés pour l'analyse des émotions à partir de voix humaines. Ces ensembles contiennent des enregistrements audio de personnes exprimant différentes émotions et sont fréquemment utilisés dans les travaux de reconnaissance des émotions basées sur la voix.

### 2.1.1 CREMA-D Dataset

Le CREMA-D (Crowd-sourced Emotional Multimodal Actors Dataset) est un dataset multimodal qui contient des enregistrements audio de 91 acteurs (hommes et femmes) exprimant 7 émotions différentes : colère, dégoût, peur, joie, tristesse, surprise et neutre. Les données sont collectées dans un environnement contrôlé, avec des acteurs qui suivent un script spécifique. Ce dataset est particulièrement utile pour la reconnaissance émotionnelle en raison de la diversité des expressions émotionnelles et des variétés de tonalités vocales présentes.

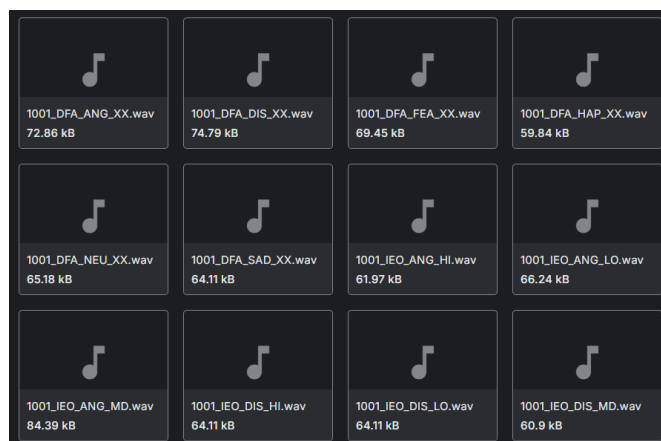


FIGURE 2.1 – Distribution des émotions dans le dataset CREMA-D

### 2.1.2 TESS Dataset

Le dataset TESS (Toronto emotional speech set) est composé de 2000 enregistrements audio dans lesquels 2 actrices expriment 7 émotions : joie, tristesse, colère, peur, surprise, dégoût, et

neutre. Ce dataset est particulièrement précieux pour les études sur les émotions vocales en raison de la spécificité des acteurs et de la diversité des émotions exprimées, avec une haute qualité sonore, ce qui le rend bien adapté pour l'analyse acoustique.

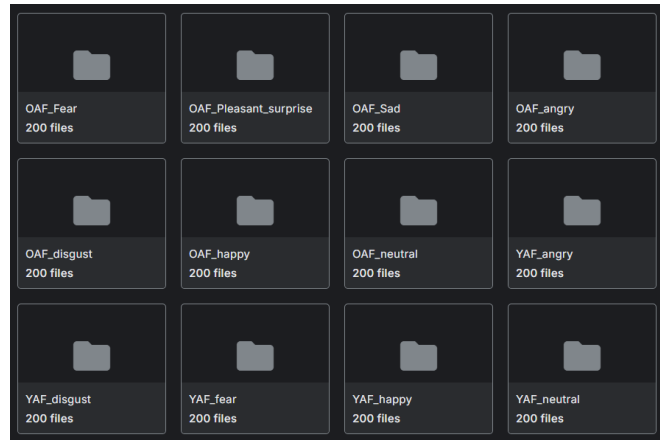


FIGURE 2.2 – Exemples d'enregistrements audio du dataset TESS

### 2.1.3 SAVEE Dataset

Le dataset SAVEE (Surrey Audio-Visual Expressed Emotion) est constitué d'enregistrements audio de 4 hommes et 3 femmes, exprimant 7 émotions : joie, tristesse, colère, peur, surprise, dégoût, et neutre. Les enregistrements sont réalisés dans un environnement contrôlé, et le dataset se distingue par sa petite taille et son focus sur des émotions naturelles. Bien que plus petit que d'autres datasets, le SAVEE est utilisé pour des recherches spécifiques dans l'analyse des émotions vocales.

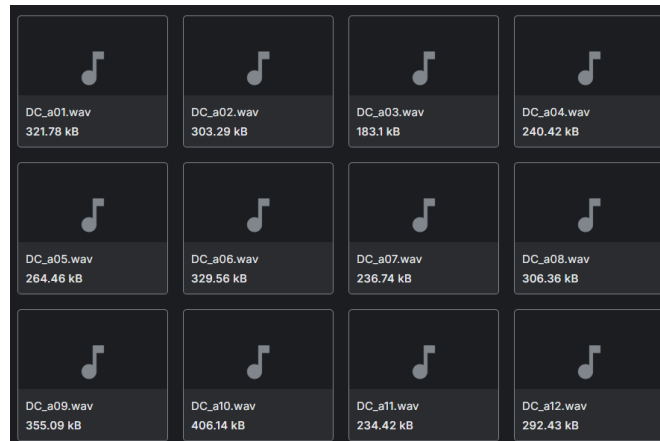


FIGURE 2.3 – Exemples d'enregistrements du dataset SAVEE

Ces trois datasets sont largement utilisés dans les recherches sur la reconnaissance des émotions, en particulier dans les domaines du traitement du langage naturel et de l'analyse acoustique des voix.

## 2.2 Data Preprocessing

Pour exploiter les ensembles de données CREMA-D, SAVEE et TESS, une étape de prétraitement est nécessaire. Cette section décrit les étapes suivies pour unifier et préparer les données pour l'analyse.

### 2.2.1 Prétraitement des données SAVEE

Les fichiers audio du dataset SAVEE sont situés dans le dossier **SAVEE/ALL**. Chaque fichier suit un schéma de dénomination permettant d'identifier l'émotion associée à l'enregistrement. Un mappage entre les codes d'émotions (*a, d, f, h, sa, su, n*) et leurs libellés (*anger, disgust, fear, happiness, sadness, surprise, neutral*) a été défini. Les étapes principales du prétraitement sont les suivantes :

- Lecture des fichiers avec l'extension **.wav**.
- Extraction du code d'émotion à partir du nom de fichier.
- Conversion des codes en libellés d'émotions grâce à un dictionnaire de mappage.
- Création d'un *DataFrame* Pandas contenant les chemins d'accès absolus et les étiquettes des fichiers.

### 2.2.2 Prétraitement des données TESS

Les fichiers audio du dataset TESS sont situés dans des sous-dossiers organisés par émotion dans **TESS/TESS Toronto emotional speech set data**. Les sous-dossiers contiennent les fichiers **.wav** correspondant à chaque émotion. Les étapes principales du prétraitement incluent :

- Parcours des sous-dossiers pour identifier les fichiers audio.
- Extraction des émotions à partir des noms des sous-dossiers et conversion à l'aide d'un mappage.
- Création d'un *DataFrame* Pandas avec les chemins d'accès absolus et les étiquettes.

### 2.2.3 Prétraitement des données CREMA-D

Les fichiers audio du dataset CREMA-D sont situés dans **cremad/AudioWAV**. Les étiquettes d'émotion (*ANG, DIS, FEA, HAP, NEU, SAD*) sont extraites des noms des fichiers. Les étapes du prétraitement incluent :

- Identification des fichiers audio au format **.wav**.
- Extraction des codes d'émotion et mappage vers leurs libellés correspondants.
- Création d'un *DataFrame* Pandas listant les chemins d'accès et les étiquettes.

### 2.2.4 Fusion des ensembles de données

Une fois les *DataFrames* individuels générés pour chaque dataset, ils sont fusionnés en un seul *DataFrame* contenant toutes les données :

- Les *DataFrames* CREMA-D, SAVEE et TESS sont concaténés en une seule table.
- Les indices sont réinitialisés pour garantir la continuité.

Cette étape permet de centraliser les données pour des tâches ultérieures telles que l'analyse ou la modélisation.

### 2.2.5 Visualisation des données

Pour mieux comprendre la distribution des émotions dans l'ensemble fusionné, une visualisation graphique est réalisée. Un diagramme en barres est généré pour montrer le nombre d'échantillons par émotion. Cette étape met en évidence tout déséquilibre éventuel dans les classes émotionnelles, essentiel pour la phase de modélisation.

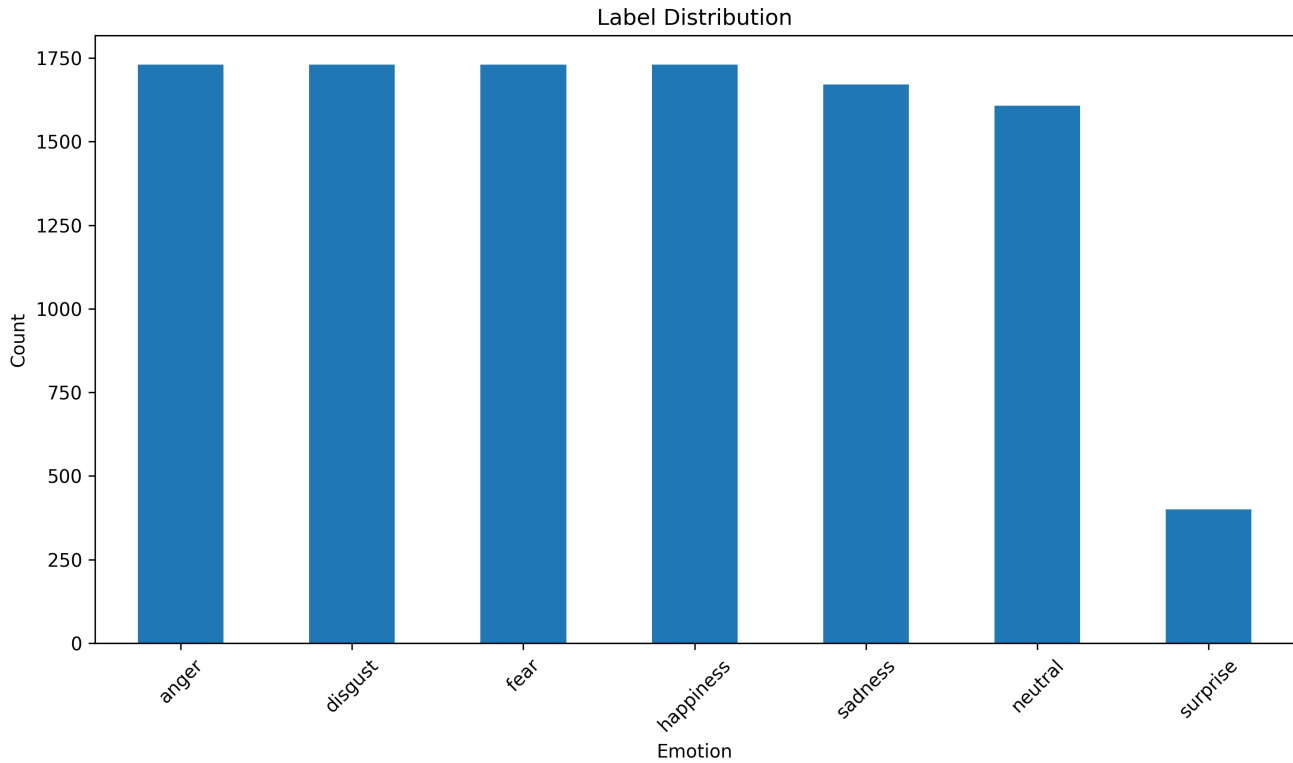


FIGURE 2.4 – Distribution des émotions dans l'ensemble fusionné

### 2.2.6 Exportation des données

Les données prétraitées sont exportées dans un fichier CSV (`file_paths_labels.csv`) pour une utilisation ultérieure dans les étapes de construction des modèles.

## 2.3 Feature Extraction

La phase de l'extraction des caractéristiques constitue une étape essentielle dans le traitement des données audio pour les tâches d'apprentissage automatique. Elle consiste à convertir les signaux audio bruts en représentations numériques significatives, encapsulant les propriétés essentielles de l'audio. Cette section décrit les principales techniques utilisées pour l'extraction des caractéristiques dans notre projet.

### 2.3.1 Taux d'Échantillonnage

Le taux d'échantillonnage ou fréquence d'échantillonnage fait référence au nombre d'échantillons d'audio prélevés par seconde lors du processus de conversion analogique-numérique. C'est un paramètre fondamental qui décrit la représentation numérique d'un signal audio. Il influence à la fois la qualité audio et la taille du fichier résultant.

#### Les Taux d'Échantillonnage dans notre Jeu de Données

Notre jeu de données contient des fichiers audio avec **quatre différents taux d'échantillonnage**, ce qui reflète la diversité des sources d'enregistrement. Ces taux sont les suivants :

- **8 kHz** : Ce taux est couramment utilisé dans les applications de téléphonie, où une bande passante limitée est suffisante pour la voix humaine. Il offre une qualité basique mais adéquate pour la compréhension.
- **16 kHz** : Utilisé dans des applications comme la reconnaissance vocale, il offre une meilleure résolution sonore que 8 kHz, tout en maintenant un compromis acceptable entre la qualité et la taille du fichier.
- **22.05 kHz** : Ce taux est fréquemment utilisé pour les enregistrements audio à usage général où une qualité audio modérée est suffisante.
- **44.1 kHz** : Standard pour l'audio de haute qualité, ce taux est utilisé dans les applications de musique et d'enregistrement, offrant une reproduction sonore fidèle.

Ces variations dans les taux d'échantillonnage nécessitent une normalisation ou une gestion appropriée lors de la phase de traitement des données pour garantir une cohérence et une comparabilité entre les fichiers audio. Dans notre projet, les fichiers ont été rééchantillonnés à un taux commun pour une analyse homogène et efficace.

### 2.3.2 Techniques d'Augmentation des Données

Pour améliorer la robustesse du modèle, deux techniques d'augmentation des données ont été utilisées :

- **Injection de Bruit** : Ajoute un bruit aléatoire au signal audio pour simuler des environnements d'enregistrement variés.
- **Étirement Temporel** : Modifie la vitesse de lecture de l'audio sans en changer la hauteur.

### 2.3.3 Techniques d'Extraction des Caractéristiques

Les principales caractéristiques ont été extraites des fichiers audio à l'aide de la bibliothèque *Librosa* :

- **Taux de Passage à Zéro (ZCR)** : Mesure le taux de changement de signe dans le signal, corrélé avec le niveau de bruit.



- **Chroma STFT (Short-Time Fourier Transform)** : Capture les classes de hauteur (par exemple, les notes dans une gamme musicale) en fonction du temps.
- **MFCC** : Fournit une représentation compacte du spectre de puissance du signal.
- **RMS (Root Mean Square)** : Quantifie l'amplitude du signal.
- **Spectrogramme Mel** : Un spectrogramme mis à l'échelle logarithmique basé sur l'échelle Mel pour les bandes de fréquence.
- **Contraste Spectral** : Met en évidence la différence entre les pics et les creux dans le spectre, utile pour les signaux rythmiques et tonaux.
- **Tonnetz (Tonal Centroid Features)** : Représente les relations tonales telles que les structures harmoniques ou mélodiques.

### 2.3.4 Création des Vecteurs de Caractéristiques

Les caractéristiques extraites ont été concaténées dans un seul vecteur de caractéristiques pour chaque fichier audio. Ce vecteur représente de manière complète le signal audio, incluant ses caractéristiques temporelles, spectrales et tonales. Les vecteurs de caractéristiques ont été sauvegardés dans un fichier CSV pour un traitement ultérieur.

Grâce à la combinaison de ces différentes caractéristiques et des techniques d'augmentation des données, nous avons préparé efficacement les données audio pour les tâches suivantes, telles que la classification des émotions ou l'analyse des sentiments.

|    | FD         | FE         | FF         | FG         | FH         | FI         | FJ         | FK         | FL         | FM         | FN          | FO          | FP          | FQ          | FR          | FS          | FT        | FU  | Label |
|----|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------|-----|-------|
| 1  | 159        | 160        | 161        | 162        | 163        | 164        | 165        | 166        | 167        | 168        | 169         | 170         | 171         | 172         | 173         | 174         | 175       | 176 | anger |
| 2  | 4.50E-05   | 5.16E-05   | 3.93E-05   | 10.3466774 | 4.45304656 | 8.66208951 | 10.3990784 | 15.5476386 | 16.2808797 | 12.0998165 | 0.00988805  | 0.01315529  | -0.03708248 | -0.05640053 | 0.00900663  | 0.00747894  | anger     |     |       |
| 3  | 1.57E-05   | 1.96E-05   | 1.76E-05   | 8.61228588 | 4.18957241 | 8.73687393 | 9.74395492 | 14.5026511 | 15.3353534 | 12.6091979 | -0.01407379 | -0.02234629 | -0.03215682 | -0.04239007 | 0.01448888  | 0.00031587  | disgust   |     |       |
| 4  | 0.00021142 | 0.00016041 | 0.00013557 | 9.6461124  | 4.74233916 | 8.57254145 | 10.498687  | 15.0303118 | 15.5852165 | 12.7470772 | -0.00879472 | 0.03637496  | -0.02776894 | -0.03692981 | 0.01885682  | 0.00579925  | fear      |     |       |
| 5  | 6.35E-05   | 4.85E-05   | 5.26E-05   | 10.8759159 | 5.13906253 | 9.41238603 | 10.1893627 | 15.098141  | 16.1171428 | 12.5990005 | -0.00348721 | 0.02527221  | -0.0451965  | -0.03336858 | -0.00473818 | 0.00279426  | happiness |     |       |
| 6  | 9.48E-06   | 1.24E-05   | 1.07E-05   | 9.23423117 | 4.55958204 | 8.80493335 | 9.82141851 | 14.956421  | 16.4533915 | 12.5159492 | 0.01856142  | 0.00500206  | -0.02516578 | -0.03676902 | 0.01082132  | 0.00642049  | neutral   |     |       |
| 7  | 1.55E-05   | 2.18E-05   | 1.18E-05   | 9.83866242 | 4.94150026 | 8.85176758 | 9.91846131 | 14.058928  | 15.8118541 | 12.6551603 | -0.03180117 | 0.01229304  | -0.03481932 | 0.00645276  | 0.01465441  | -0.00627312 | sadness   |     |       |
| 8  | 0.0002475  | 0.00017465 | 7.99E-05   | 10.6068021 | 4.09782966 | 8.74135549 | 11.4018164 | 14.2606009 | 18.4169653 | 12.1994949 | 0.00723938  | -0.00459581 | -0.04062221 | -0.02573708 | 0.0076061   | -0.00537411 | anger     |     |       |
| 9  | 7.02E-06   | 7.08E-06   | 3.29E-06   | 9.20925656 | 4.46479766 | 8.7154075  | 10.2964348 | 13.5768547 | 16.0778087 | 12.7579914 | 0.01029736  | -0.00258802 | -0.05127088 | -0.0494408  | 0.02135998  | -0.00547831 | anger     |     |       |
| 10 | 9.25E-05   | 6.85E-05   | 6.25E-05   | 9.43153257 | 4.55279404 | 8.58666023 | 11.0849083 | 13.5695812 | 17.2515789 | 11.9500771 | -0.00129736 | 0.01137988  | -0.0822071  | -0.02198502 | 0.0189915   | 0.00863343  | anger     |     |       |
| 11 | 9.00E-05   | 6.51E-05   | 6.70E-05   | 9.79464311 | 4.57640635 | 8.43919316 | 10.6172741 | 13.8125995 | 17.289305  | 12.6153484 | -0.00382671 | -0.00082737 | -0.03541546 | 0.00532731  | 0.01446817  | -0.00319679 | disgust   |     |       |
| 12 | 2.16E-06   | 2.41E-06   | 1.47E-06   | 8.41476937 | 4.59637515 | 7.93532658 | 9.94046446 | 12.5846732 | 14.6189704 | 11.9784079 | 0.00254011  | -0.00544491 | -0.06239835 | -0.01673079 | 0.022816    | 0.00377008  | disgust   |     |       |
| 13 | 7.96E-05   | 6.08E-05   | 3.24E-05   | 9.30299836 | 4.13078635 | 7.5654191  | 10.2636243 | 13.4376954 | 15.5666502 | 12.1098755 | -0.00829279 | 0.00991252  | -0.06072415 | -0.01975984 | 0.02530899  | 0.00674515  | disgust   |     |       |
| 14 | 0.00012726 | 0.00011463 | 9.61E-05   | 10.1859137 | 4.75086347 | 8.1109483  | 10.4972929 | 14.0501114 | 16.5437391 | 12.9950486 | -0.00422092 | 0.00750706  | -0.05247927 | 0.01099733  | 0.02326059  | -0.00947675 | fear      |     |       |
| 15 | 3.91E-05   | 5.54E-05   | 4.10E-05   | 10.9397871 | 4.74305814 | 8.11059101 | 10.1537614 | 13.5942779 | 15.924934  | 11.9905389 | 0.01200745  | -0.00534985 | -0.04297479 | -0.01706571 | 0.00585033  | 0.01790575  | fear      |     |       |
| 16 | 4.49E-05   | 4.82E-05   | 2.87E-05   | 10.8116544 | 4.3652133  | 7.80045796 | 10.0002632 | 13.1682688 | 16.1832712 | 12.1297266 | 0.01313804  | 0.00801492  | -0.04062354 | -0.02643812 | 0.0182434   | 0.01808207  | fear      |     |       |
| 17 | 0.00019637 | 0.00027954 | 0.00025142 | 10.2251904 | 4.67527362 | 9.8244447  | 11.8670095 | 14.4890049 | 17.8718743 | 13.5815067 | 0.00344968  | 0.01645796  | -0.06196027 | -0.06260469 | -0.00717854 | -0.0059948  | happiness |     |       |
| 18 | 2.98E-05   | 6.13E-05   | 4.28E-05   | 8.52500764 | 4.05781692 | 8.87904132 | 9.7486129  | 12.8116418 | 16.0590124 | 12.4822032 | -0.01828241 | 0.00704121  | -0.04677025 | 0.0194732   | 0.00633313  | -0.00214961 | happiness |     |       |
| 19 | 0.00012151 | 0.0001318  | 0.00011195 | 10.0154769 | 4.68691836 | 7.88970653 | 10.3422094 | 12.8655099 | 16.4731046 | 12.3267821 | 0.01019075  | -0.02238408 | -0.02015655 | -0.01217557 | -0.00157863 | 0.00324946  | happiness |     |       |
| 20 | 3.32E-05   | 4.68E-05   | 4.17E-05   | 8.9917124  | 4.43443338 | 8.96014756 | 10.3091271 | 14.2893969 | 16.7514309 | 12.5244352 | 0.02194894  | 0.0175073   | -0.02959866 | -0.00651178 | 0.02223996  | 0.00452711  | neutral   |     |       |
| 21 | 3.46E-05   | 4.03E-05   | 1.45E-05   | 8.97522338 | 4.55901165 | 8.49172775 | 10.3574408 | 13.824337  | 14.9136838 | 11.8070033 | -0.01623475 | 0.01926968  | -0.06774306 | -0.03579818 | 0.03309789  | 0.0001665   | sadness   |     |       |
| 22 | 4.44E-05   | 3.70E-05   | 2.39E-05   | 8.77347289 | 4.44440674 | 8.6671876  | 9.89825978 | 13.4786379 | 14.8499137 | 11.8562344 | 0.00327926  | 0.02151113  | -0.0280101  | -0.01907686 | 0.03048521  | 0.0097701   | sadness   |     |       |
| 23 | 7.13E-06   | 1.32E-05   | 8.18E-06   | 8.17538672 | 4.01590744 | 8.21934815 | 10.3996507 | 14.029618  | 14.9699922 | 11.6022367 | 0.01188976  | 0.00343608  | -0.07687991 | -0.01206276 | 0.02992776  | 0.01950189  | sadness   |     |       |
| 24 | 9.55E-06   | 1.23E-05   | 6.69E-06   | 10.7799499 | 4.36256885 | 9.36984808 | 10.1003147 | 15.535772  | 17.5625016 | 12.3165282 | -0.01738295 | -0.00844256 | -0.00090185 | -0.03747147 | 0.02109387  | 0.00459115  | anger     |     |       |
| 25 | 1.32E-06   | 1.83E-06   | 1.68E-06   | 8.78954375 | 4.51515782 | 9.17989208 | 10.5184502 | 14.4933238 | 15.6628726 | 12.7659584 | 0.00207518  | 0.02336666  | -0.01150598 | 0.00048429  | 0.00426336  | 0.00191558  | disgust   |     |       |

FIGURE 2.5 – Table des features + label.

| Feature            | Start Index | End Index | Length |
|--------------------|-------------|-----------|--------|
| Zero Crossing Rate | 0           | 0         | 1      |
| Chroma STFT        | 1           | 12        | 12     |
| MFCC               | 13          | 32        | 20     |
| RMS                | 33          | 33        | 1      |
| Mel Spectrogram    | 34          | 161       | 128    |
| Spectral Contrast  | 162         | 168       | 7      |
| Tonnetz            | 169         | 174       | 6      |

FIGURE 2.6 – Index de chaque type de features.

## 2.4 Construction des Modèles

Dans cette section, nous expliquons la mise en place du modèle de machine learning, en détaillant les méthodologies et les étapes suivies pour construire et optimiser le modèle.

### 2.4.1 Validation Croisée

Pour garantir une évaluation robuste, nous avons utilisé la validation croisée à 5 plis (5-fold cross-validation). Ce procédé divise les données en cinq sous-ensembles, chacun étant à tour de rôle utilisé comme ensemble de test tandis que les autres servent à l'entraînement. Cette approche permet d'obtenir des estimations fiables de la performance et de minimiser le risque de surapprentissage.

### 2.4.2 Modèle Utilisé : Régression Logistique

Le modèle principal employé est la régression logistique, qui est bien adaptée pour les problèmes multi-classes. Nous avons exploré différentes configurations d'hyperparamètres à l'aide d'une grille de recherche, notamment :

- **Type de régularisation** :  $L1$  ou  $L2$ .
- **Force de régularisation** : Valeurs de  $C$  variant entre 0.1 et 100.
- **Inclusion de l'interception** : Vrai ou Faux.
- **Solveurs** : "Algorithmes d'optimisation comme lbfgs, newton-cg et saga.
- **Ratio  $l1$**  : Paramètre mixte pour la régularisation élastique.

Les hyperparamètres optimaux ont été déterminés en fonction de la performance obtenue lors de la validation croisée.

### 2.4.3 Pipeline de Pré-traitement

Pour assurer une meilleure qualité des données d'entrée, plusieurs étapes de pré-traitement ont été effectuées :

- **Imputation** : Les valeurs manquantes ont été remplacées par la moyenne des colonnes correspondantes.
- **Standardisation** : Toutes les variables ont été standardisées pour avoir une moyenne nulle et une variance unité.
- **Réduction de dimension** : Une analyse en composantes principales (PCA) a été appliquée pour capturer 99% de la variance totale des données.

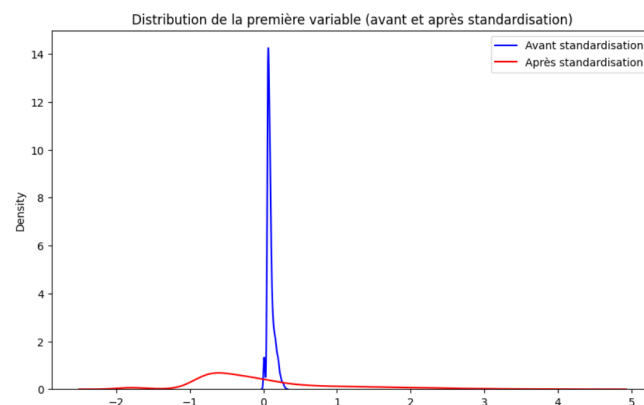


FIGURE 2.7 – Distribution de variable.

## 2.5 Évaluation des Modèles

### 2.5.1 Durée de Validation Croisée

La durée totale de la validation croisée a été enregistrée pour chaque itération et présente un indicateur clé de l'efficacité des processus de calcul. Une figure illustrant la durée totale du processus est incluse ci-dessous :

```
import time

start_time = time.time()
results = {}
for fold in fold_dfs.keys():
    print(fold)
    X_train, X_test, y_train, y_test, label_mapping = data_conversion(fold_dfs[fold])
    result = Parallel(n_jobs=num_processes)(delayed(build_and_evaluate_model)(
        model, X_train, X_test, y_train, y_test, params, fold, 'Logistic Regression') for params in param_combinations)
    results[f'fold_{fold}'] = result

end_time = time.time()
print(f"((end_time - start_time)/3600) hours")

0
1
2
3
4
4.554231847392188 hours
```

Activate Window

FIGURE 2.8 – Durée totale de validation croisée.

### 2.5.2 Optimisation des Hyperparamètres

Les configurations optimales des hyperparamètres ont été identifiées parmi un espace de recherche prédéfini. La figure ci-dessous montre les meilleurs résultats :

```
[62]: performance_metric_values = pd.DataFrame(columns=['Model', 'Fold', 'Train AUC ROC', 'Test AUC ROC'])
hyperparameters = final_error_df.loc[0, 'Hyperparameters']
print(hyperparameters)
X_train, X_test, y_train, y_test, label_mapping = data_conversion(fold_dfs[0])
model = LogisticRegression(max_iter=2000, multi_class='multinomial', solver='lbfgs')
model.set_params(**hyperparameters)
model.fit(X_train, y_train)
y_train_pred = model.predict_proba(X_train)
y_test_pred = model.predict_proba(X_test)

{'penalty': 'l1', 'C': 0.1, 'fit_intercept': False, 'solver': 'saga', 'l1_ratio': 0.5}
```

FIGURE 2.9 – Meilleurs Hyperparamètres et fitting du modèle.

### 2.5.3 Performance des Modèles

La performance des modèles a été mesurée à l'aide de l'AUC-ROC, qui reflète la capacité du modèle à distinguer entre les classes. Les résultats obtenus pour l'ensemble d'entraînement et l'ensemble de test sont récapitulés ci-dessous :

- **AUC-ROC Entraînement** : 0.869. Cela indique une bonne capacité de classification des données d'entraînement par le modèle.
- **AUC-ROC Test** : 0.873. Ce résultat montre que le modèle généralise bien sur des données non vues.

Ces valeurs proches indiquent une absence significative de surapprentissage (overfitting), ce qui reflète un bon équilibre entre biais et variance dans le modèle.

|   | Model               | Fold | Train AUC ROC | Test AUC ROC |
|---|---------------------|------|---------------|--------------|
| 0 | Logistic_Regression | 0    | 0.868985      | 0.87284      |

FIGURE 2.10 – Résultats

Pour mieux visualiser la performance du modèle, les courbes ROC pour les ensembles d’entraînement et de test sont présentées ci-dessous :

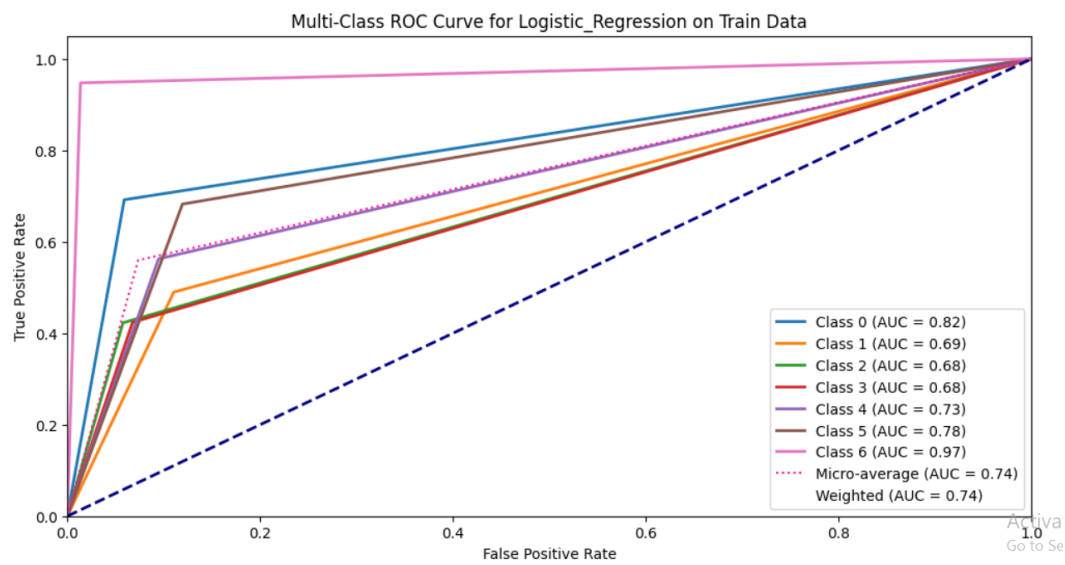


FIGURE 2.11 – Courbe ROC pour l’ensemble d’entraînement avec l’AUC associée.

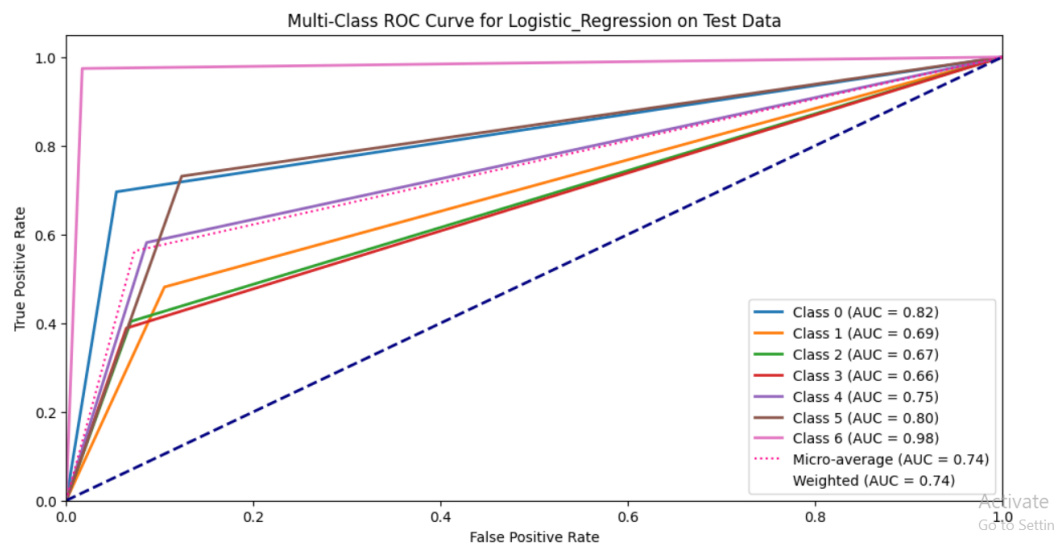


FIGURE 2.12 – Courbe ROC pour l’ensemble de test avec l’AUC associée.

Ces résultats mettent en évidence les performances du modèle dans des conditions de validation rigoureuses et permettent de valider son utilisation pour le problème à traiter.

# Deuxième partie

## Data Visualisation

# Data Preprocessing

## Les Données

Dans cette section, nous examinons la distribution des émotions dans la base de données utilisée pour notre analyse. Le graphique ci-dessous présente un histogramme qui illustre la répartition des émotions, mettant en évidence les différences de fréquence entre les classes d'émotions.

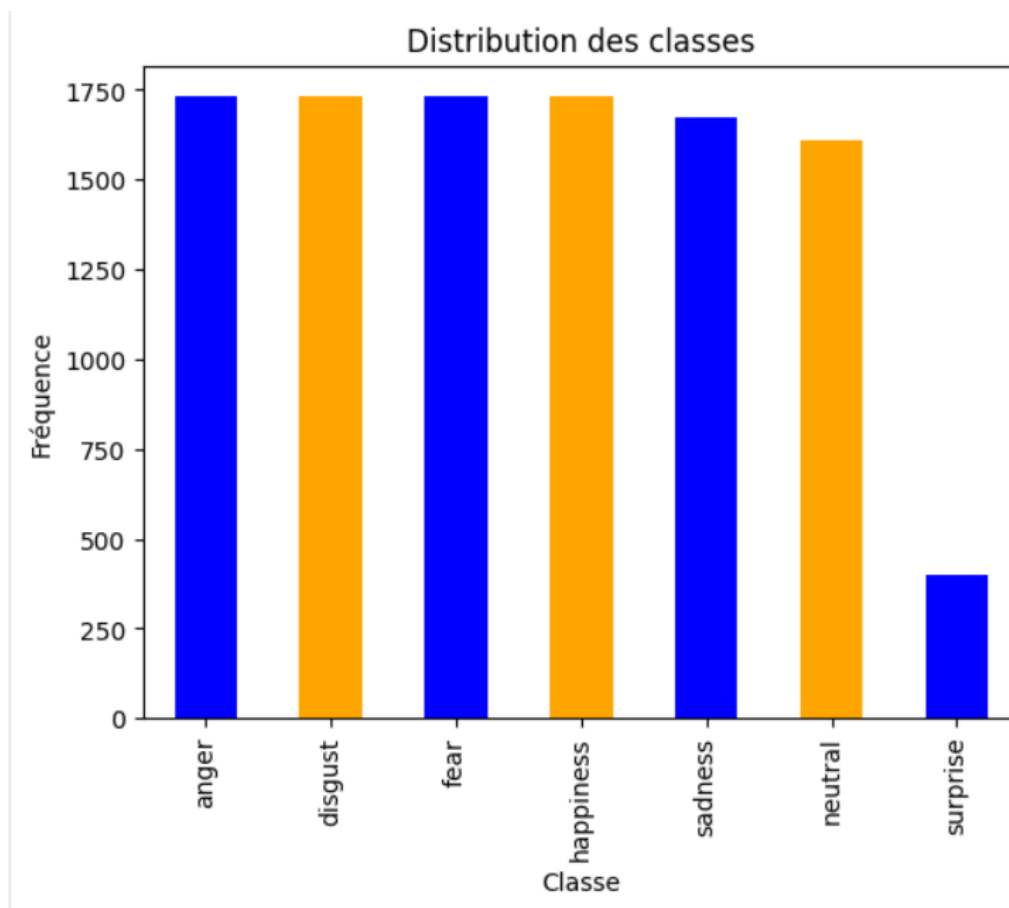


FIGURE 3.1 – Distribution des émotions dans la base de données

L'analyse de cet histogramme révèle que certaines émotions, telles que la joie et la colère, sont sur-représentées, tandis que d'autres, comme la surprise, sont sous-représentées. Cela peut

introduire un déséquilibre dans les modèles de classification et influencer leurs performances. Il est important de considérer ces disparités dans le traitement et l'évaluation des données.

## Standardisation des Données

Afin de garantir que toutes les variables ont une échelle comparable et d'éviter que certaines caractéristiques dominent les autres, nous avons standardisé les données. Le graphique ci-dessous montre l'impact de cette standardisation sur les variables de notre jeu de données.

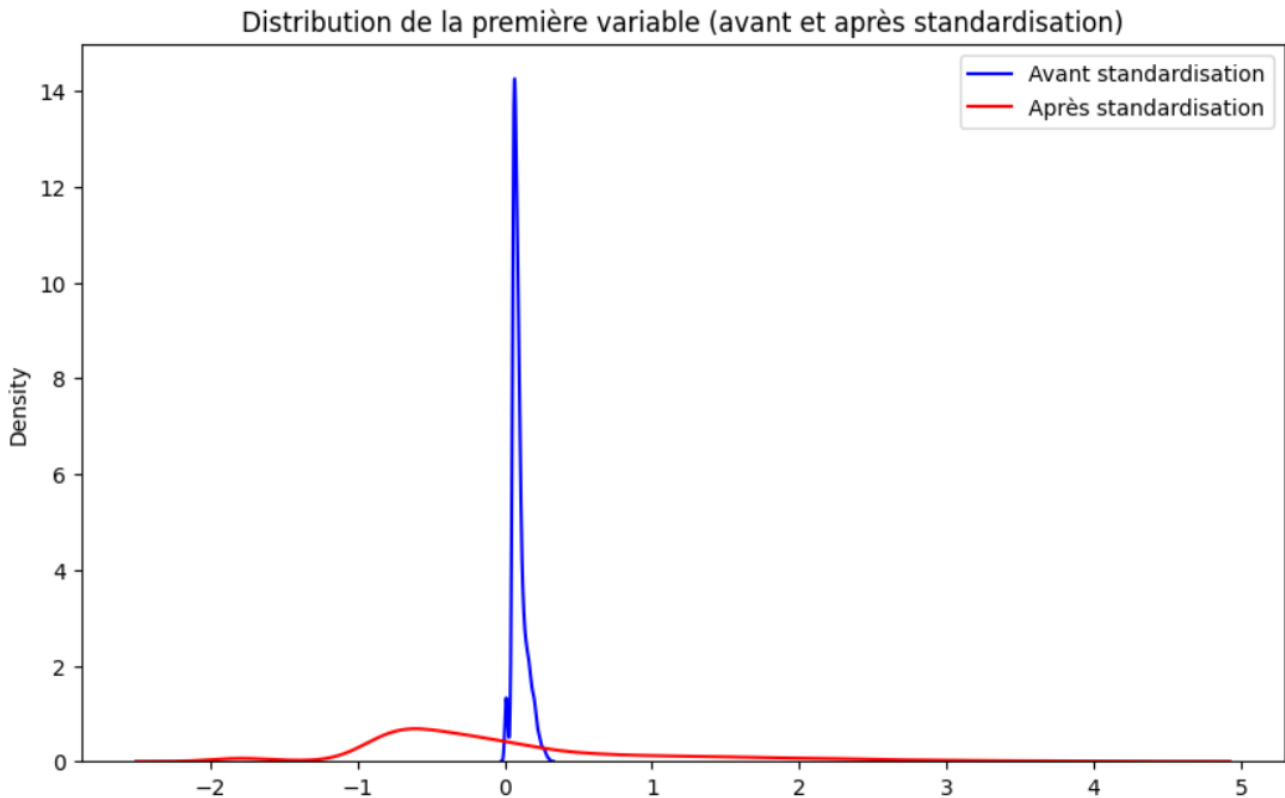


FIGURE 3.2 – Impact de la standardisation des données

L'observation de ce graphique révèle que la standardisation a permis de centrer et réduire les données, rendant chaque caractéristique comparable et facilitant ainsi les étapes suivantes de notre analyse.

## Analyse en Composantes Principales (PCA)

Pour réduire la dimensionnalité de nos données tout en préservant le maximum d'information possible, nous avons appliqué l'Analyse en Composantes Principales (PCA). Le graphique ci-dessous montre la répartition des données après application de la PCA, mettant en lumière les composantes principales qui expliquent la variance des données.

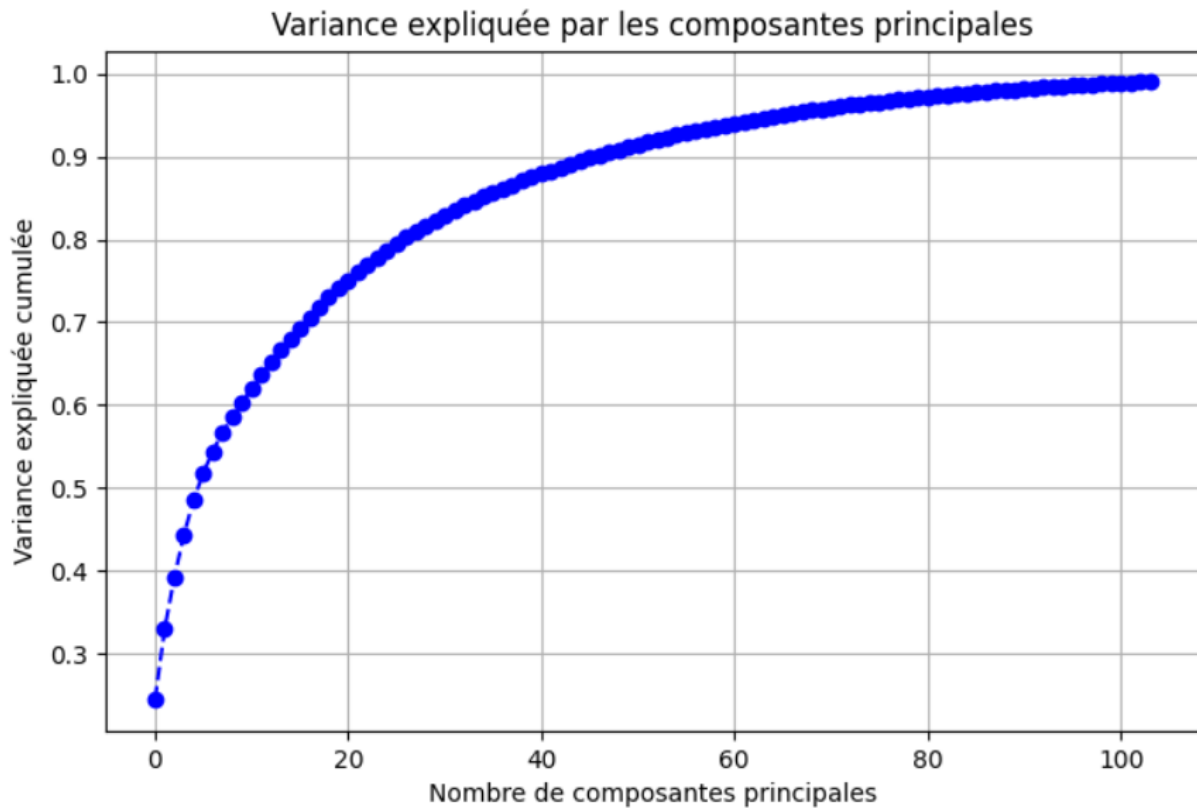


FIGURE 3.3 – Répartition des données après PCA

Les résultats de la PCA montrent que les premières composantes principales capturent une part importante de la variance des données, permettant ainsi de réduire la dimensionnalité tout en conservant l'information essentielle pour les étapes ultérieures de notre analyse.



## Audio Features

Cette section explore les caractéristiques audio extraites à partir des signaux, notamment les coefficients cepstraux fréquentiels de Mel (MFCC), le spectrogramme de Mel, le Zero Crossing Rate (ZCR), et le spectrogramme STFT. Ces caractéristiques sont essentielles pour capturer les éléments acoustiques pertinents des données.

### 4.1 Waveplot

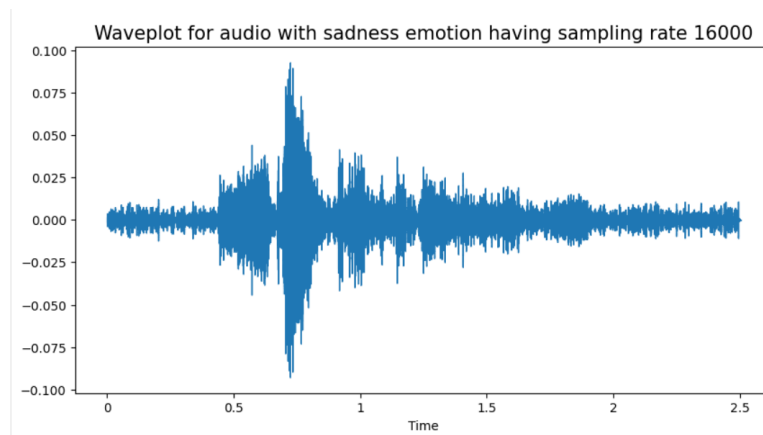


FIGURE 4.1 – Waveplot pour une émotion donnée

Le waveplot montre l'évolution de l'amplitude du signal audio dans le temps. Il permet de visualiser directement l'énergie et les variations du signal correspondant à une émotion spécifique.

## 4.2 Spectrogramme STFT

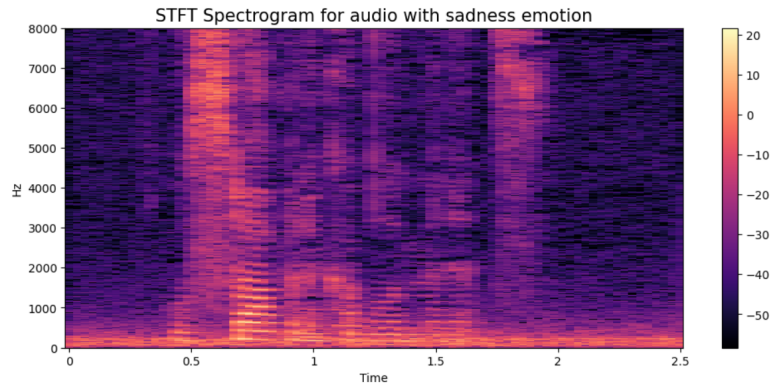


FIGURE 4.2 – Spectrogramme STFT pour une émotion donnée

Le spectrogramme STFT présente la distribution spectrale du signal en fonction du temps. Les zones de forte intensité indiquent des fréquences caractéristiques associées à des émotions spécifiques.

## 4.3 MFCC

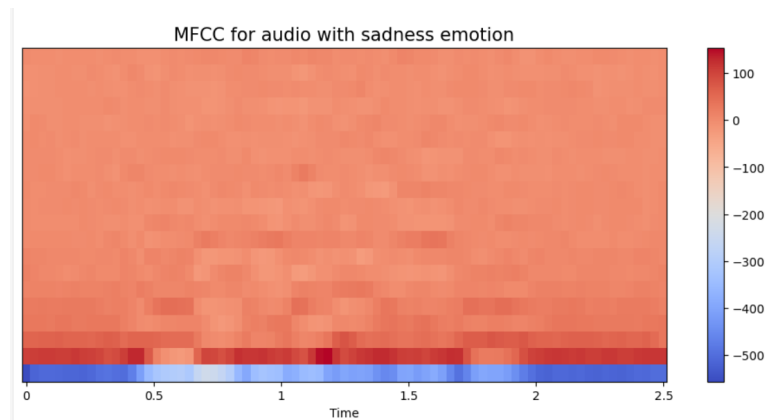


FIGURE 4.3 – Coefficients MFCC pour une émotion donnée

Les MFCC capturent les caractéristiques perceptuelles importantes des signaux audio. Les variations dans ces coefficients mettent en évidence les aspects acoustiques qui diffèrent entre les émotions.

## 4.4 Spectrogramme de Mel

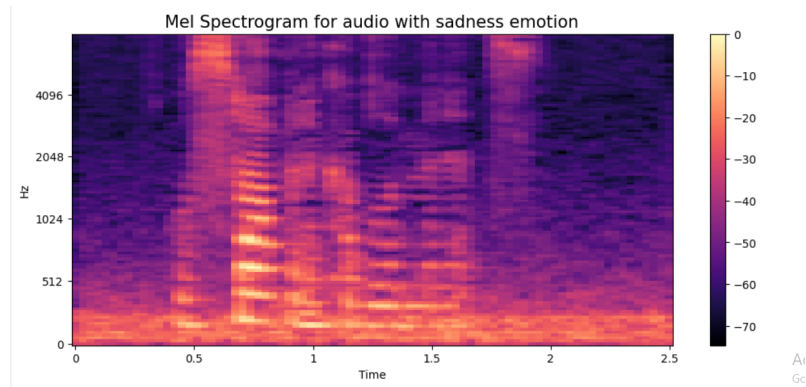


FIGURE 4.4 – Spectrogramme de Mel pour une émotion donnée

Le spectrogramme de Mel montre la distribution énergétique des fréquences dans une échelle perceptive. Cette représentation met en avant les fréquences dominantes qui influencent la reconnaissance des émotions.

## 4.5 Zero Crossing Rate (ZCR)

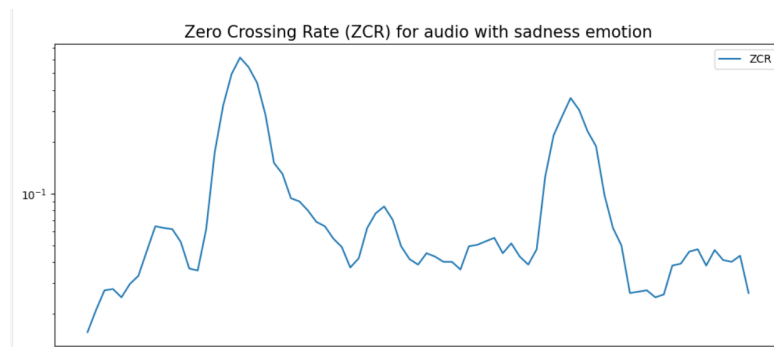


FIGURE 4.5 – Taux de passage à zéro (ZCR) pour une émotion donnée

Le ZCR mesure la fréquence à laquelle le signal audio traverse l'axe zéro. Les valeurs plus élevées ou plus faibles du ZCR sont souvent associées à des variations spécifiques dans le contenu émotionnel.

En conclusion, ces visualisations permettent une analyse approfondie des signaux audio et mettent en lumière les caractéristiques déterminantes pour les modèles de classification émotionnelle.

## Le Modèle

Cette section présente les performances du modèle d'analyse des émotions basé sur l'audio et le texte. Les courbes ROC obtenues sur les ensembles de données d'entraînement et de test illustrent la capacité du modèle à distinguer les classes d'émotions.

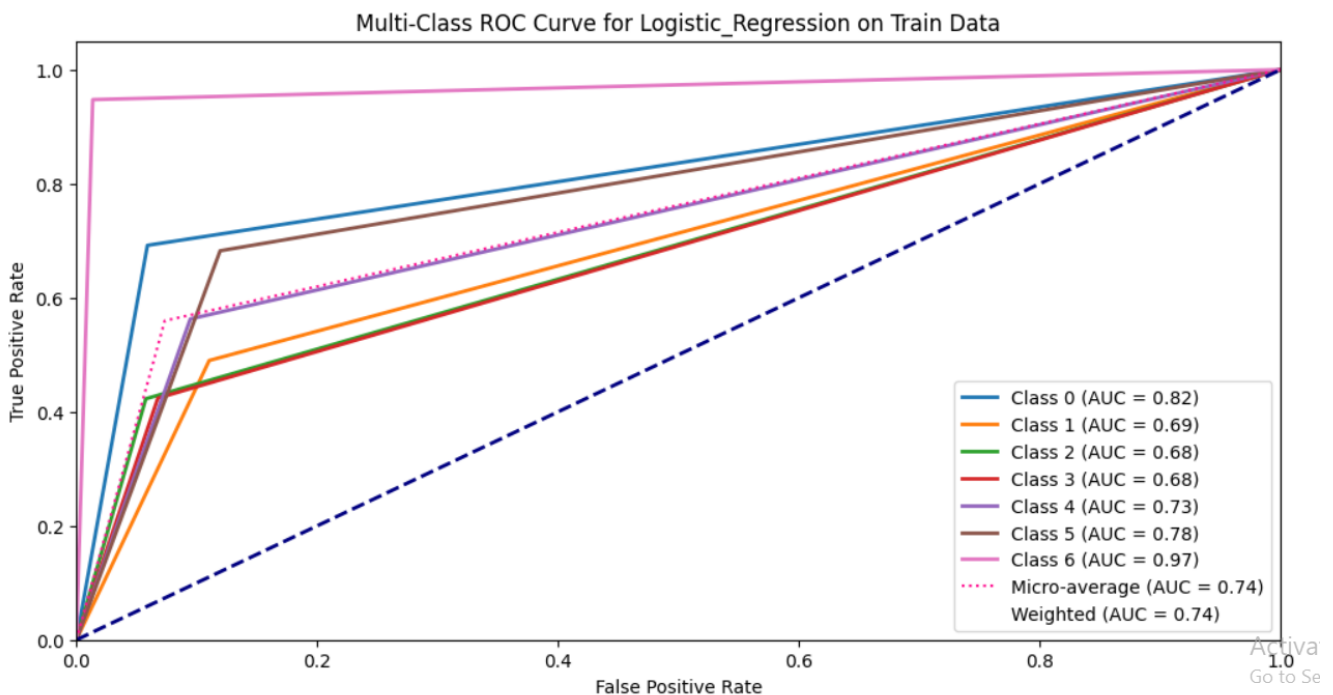


FIGURE 5.1 – Courbe ROC pour l'ensemble d'entraînement

La figure 5.1 montre une excellente performance sur l'ensemble d'entraînement, avec une aire sous la courbe (AUC) proche de 1. Cela reflète une bonne aptitude du modèle à s'adapter aux données vues pendant l'entraînement.

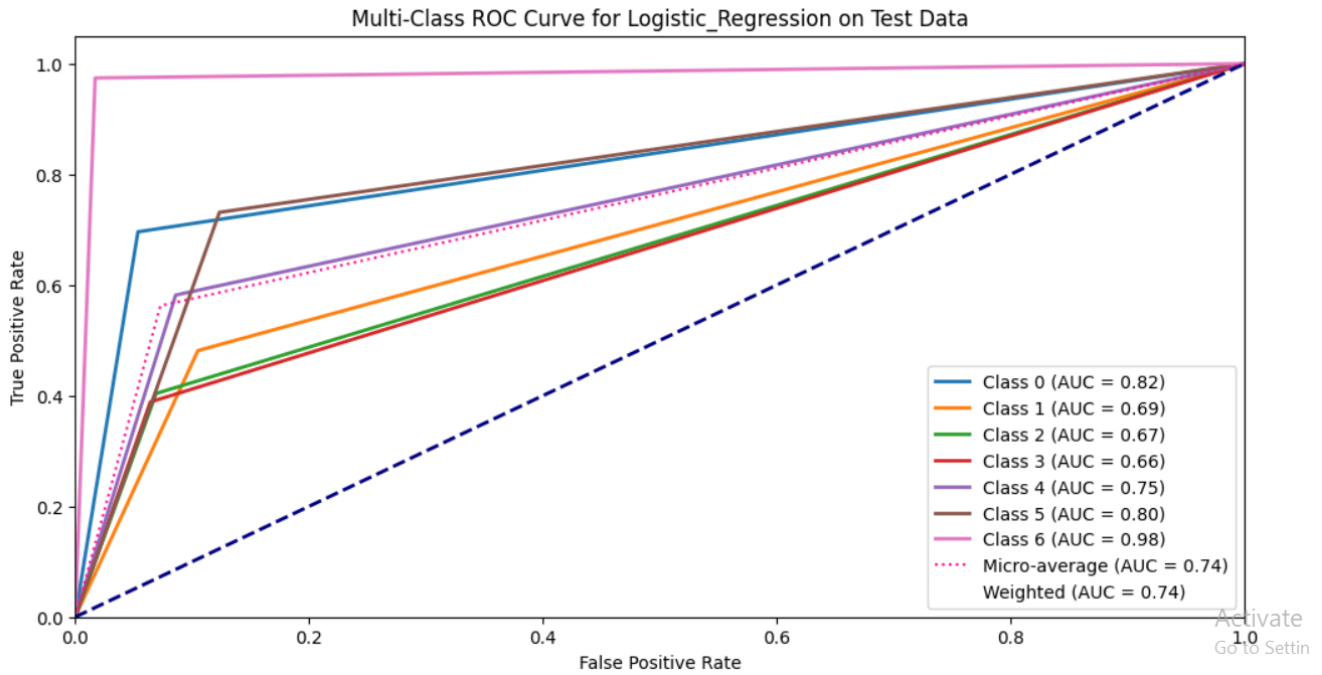


FIGURE 5.2 – Courbe ROC pour l'ensemble de test

Dans la figure 5.2, la courbe ROC de l'ensemble de test reste également élevée, bien que l'AUC soit légèrement inférieure à celle de l'entraînement. Cela indique que le modèle généralise bien sur les nouvelles données, tout en étant légèrement affecté par des émotions moins représentées.

En conclusion, ces visualisations fournissent une vue complète des données, des caractéristiques audio, et des performances du modèle, facilitant une meilleure compréhension et amélioration des résultats obtenus.

# Troisième partie

## Devops

# Chapitre 6

## Méthode Agile

Pour la gestion de notre projet, nous avons appliqué la méthode **Agile** pour organiser et structurer les tâches. Plus précisément, nous avons utilisé la méthode **Kanban**, qui permet de gérer efficacement le flux de travail en divisant les tâches en petites unités et en suivant leur progression.

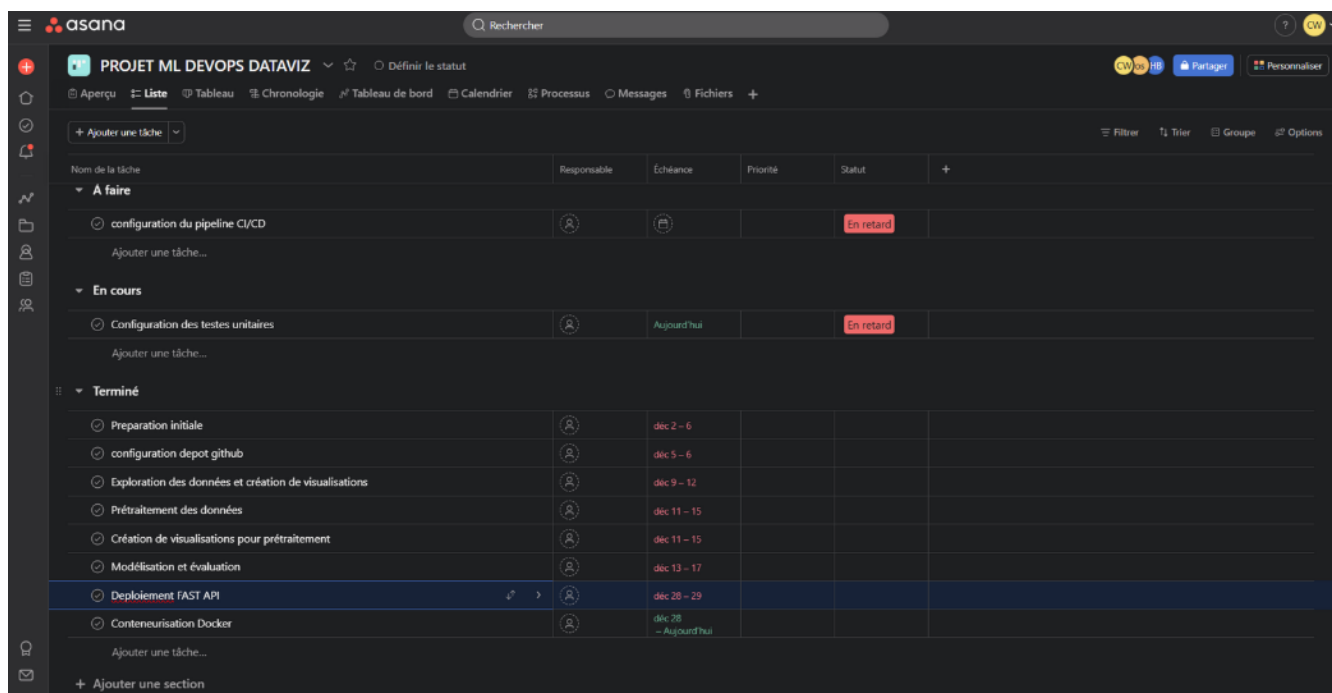


FIGURE 6.1 – Méthode agile : Kanban

## Chapitre 7

# API avec FastAPI

Nous avons déployé une API en utilisant **FastAPI**, un framework léger et rapide pour créer des APIs modernes. L'API a été conçue de manière simple, offrant une interface accessible et facile à utiliser pour interagir avec le modèle de classification des émotions. La figure 7.1 montre l'interface de l'API.

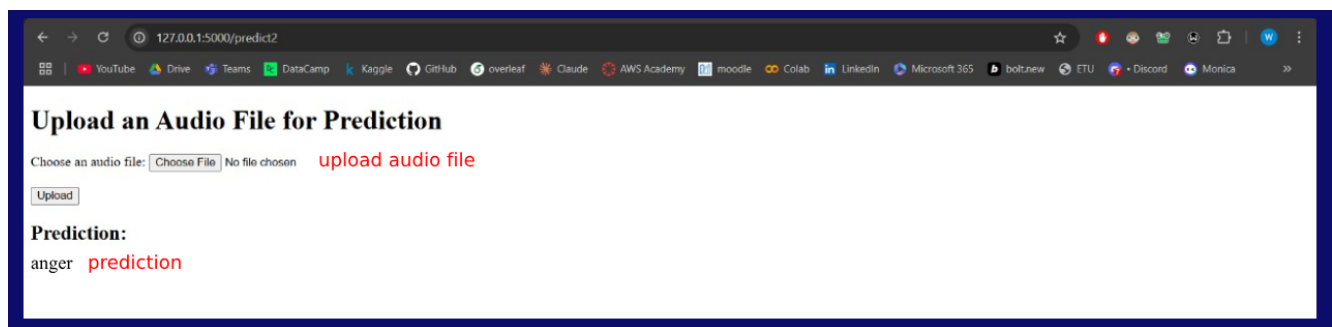


FIGURE 7.1 – API avec FastAPI



# Docker

Afin de garantir la portabilité et la reproductibilité de notre environnement, nous avons conteneurisé l'API avec **Docker**. La première étape de ce processus a consisté à créer un fichier Docker. Ce fichier permet de définir les dépendances nécessaires à l'exécution de l'API dans un environnement isolé. La figure 8.1 montre le contenu du fichier Docker.

```
notebooks > Dockerfile > ...
1  # Use the official Python image as the base image
2  FROM python:3.10-slim
3
4  # Set the working directory inside the container
5  WORKDIR /app
6
7  # Copy the requirements.txt file into the container
8  COPY requirements.txt /app/
9
10 # Install the dependencies
11 RUN pip install --no-cache-dir -r requirements.txt
12
13 # Copy the entire application code into the container
14 COPY . /app/
15
16 # Copy the .pkl model file into the container
17 COPY mymodel.pkl /app/
18
19 # Expose port 5000 for the Flask application
20 EXPOSE 5000
21
22 # Set the environment variable to disable Python buffering for logs
23 ENV PYTHONUNBUFFERED=1
24
25 # Run the Flask application when the container starts
26 CMD ["python", "app.py"]
27
```

FIGURE 8.1 – Fichier Docker

La figure 8.2 présente le fichier **requirements.txt** qui liste toutes les dépendances nécessaires pour faire fonctionner l'API et le modèle de classification des émotions.

```

notebooks > requirements.txt
1  Flask==2.2.2
2  joblib==1.4.2
3  librosa==0.10.2.post1
4  numpy==2.0.2
5  scikit-learn==1.6.0
6  werkzeug==2.2.2

```

FIGURE 8.2 – Fichier requirements.txt

Une fois le fichier Docker créé, nous avons procédé à la construction de l'image Docker. Cette étape permet de créer une image exécutable contenant tous les éléments nécessaires à l'exécution de l'API. La figure 8.3 illustre cette étape de construction.

```

C:\Users\wiamc\OneDrive\Desktop\ml_project\gd-project-swh\notebooks>docker build -t projetdevops .
[+] Building 2.2s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 655B
=> [internal] load metadata for docker.io/library/python:3.10-slim
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.10-slim@sha256:bdc6c5b8f725df8b009b32da65cbf46bfd24d1c86dce2e6169452c19
=> => resolve docker.io/library/python:3.10-slim@sha256:bdc6c5b8f725df8b009b32da65cbf46bfd24d1c86dce2e6169452c19
=> [internal] load build context
=> => transferring context: 1.35kB
=> CACHED [2/5] WORKDIR /app
=> CACHED [3/5] COPY requirements.txt /app/
=> CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt
=> [5/5] COPY . /app/
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:241e449f1a323fe858db64f6559e05c8479fe69f326d09f7b0a20870a1b42462
=> => exporting config sha256:9d6f47c3a29a4460283fe9c1cf206b605a2438b2a41779d2ba7ef79f7e95064e
=> => exporting attestation manifest sha256:031be443326e03f3aa33515e3d459af06c126eae4dd7210b1962722661692e78
=> => exporting manifest list sha256:087a127993a4d17434b74faf30340a898f118746a5848638b3ead0a0a01ca585
=> => naming to docker.io/library/projetdevops:latest
=> => unpacking to docker.io/library/projetdevops:latest
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/muhg5yqfytcfilnssa3wdkpsy

```

```

C:\Users\wiamc\OneDrive\Desktop\ml_project\gd-project-swh\notebooks>docker image ls
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
projetdevops         latest          087a127993a4    2 minutes ago  881MB
dockersamples/wordsmith-api    latest          7eb73732187e    3 weeks ago    738MB
dockersamples/wordsmith-web    latest          bf93899a45d4    3 weeks ago    25.8M
figlet               latest         fe83b61bd201    4 weeks ago    192MB

```

FIGURE 8.3 – Construction de l'image Docker

La figure 8.4 montre l'image Docker générée après la construction. Cette image contient l'API et toutes ses dépendances, prêtes à être exécutées dans n'importe quel environnement compatible avec Docker.

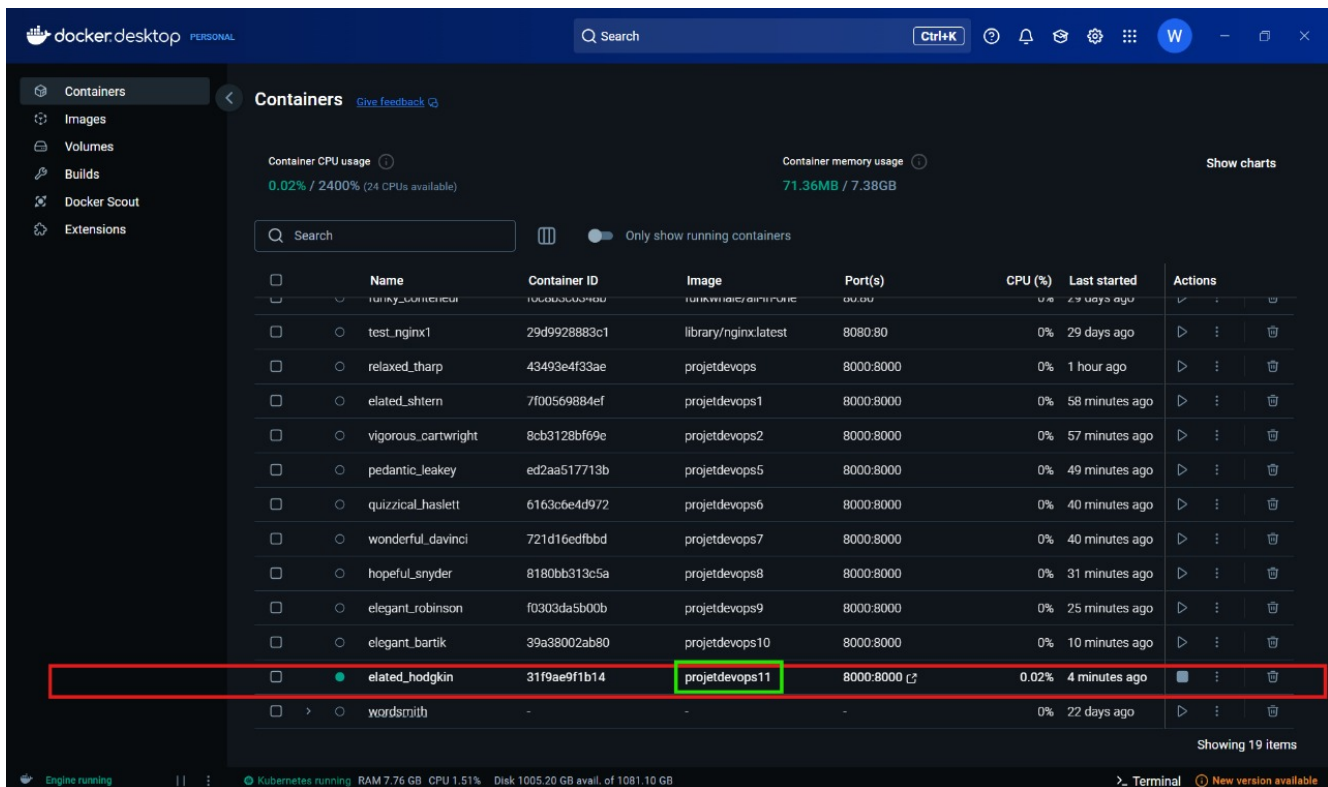


FIGURE 8.4 – Image sur Docker

Une fois l'image Docker construite, nous avons lancé le conteneur Docker pour exécuter l'API. La figure 8.5 montre ce processus de lancement, qui permet de faire fonctionner l'API dans un environnement isolé, garantissant ainsi la cohérence et la stabilité de son fonctionnement.

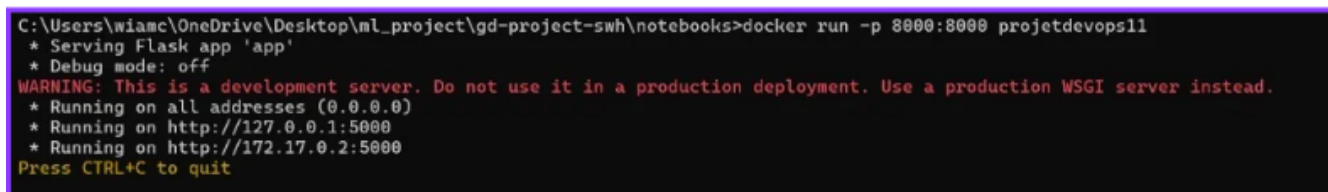


FIGURE 8.5 – Exécution de l'image Docker

En résumé, nous avons déployé une API avec notre modèle de classification des émotions en utilisant FastAPI. Ensuite, nous avons conteneurisé l'API avec Docker pour garantir sa portabilité et sa reproductibilité. Nous avons utilisé la méthode Agile, en particulier Kanban, pour organiser et gérer efficacement le projet, ce qui nous a permis de suivre et d'ajuster les tâches au fur et à mesure de leur avancement.

# Conclusion Générale

Ce projet met en évidence le rôle crucial des technologies modernes et des méthodologies avancées pour relever des défis complexes dans le domaine de l'analyse des sentiments. L'intégration de différentes disciplines, telles que le Machine Learning, la Visualisation de Données et le DevOps, a permis de développer une approche efficace et cohérente pour répondre aux objectifs fixés.

Les résultats obtenus démontrent la pertinence d'une approche méthodique et rigoureuse, tout en soulignant l'importance de la qualité des données et de l'optimisation des outils utilisés. Ce projet ouvre également la voie à des perspectives prometteuses pour des applications pratiques et des recherches futures, en explorant des solutions encore plus performantes et adaptées à des contextes variés.

En conclusion, ce travail illustre la puissance de la combinaison entre la science des données et l'ingénierie pour concevoir des solutions innovantes et évolutives répondant aux besoins actuels des utilisateurs et des entreprises.