

RECOGNAITON

Reconnaissance d'Emotions dans la Voix



*Alexandre
GASTINEL*



*Pierre
SAINCTAVIT*



*Nathan
LEWY*



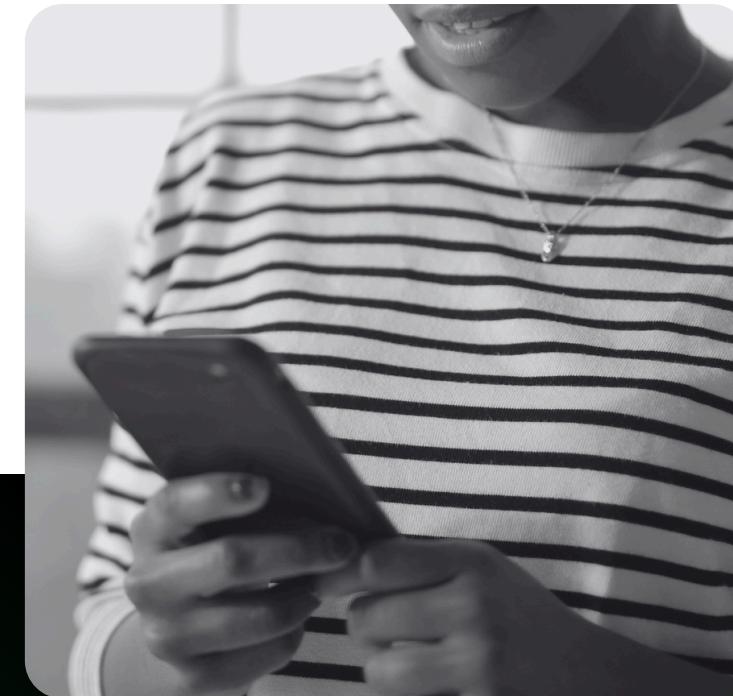
*Paul
LEMAIRE*

PLAN

- I. Introduction & Etat de l'art**
- II. Traitement de la parole: Préparation des données**
- III. Modèles Classiques & Résultats**
- IV. Réseaux Avancés & Résultats.**
- V. Conclusion**
- VI. Annexe: Courbes & Bibliographies**

INTRODUCTION

Quel Intérêt ?



Santé

Amélioration du suivi de la santé mentale et émotionnelle & Détection précoce de troubles.

Communication

Adaptation des réponses d'un chat bot, des meilleur sous-titrages

Justice

Évaluation de la sincérité et détection d'émotion de témoignages.

Marketing

Compréhension de la réaction d'un consommateur face à une publicité.

TRAITEMENT DE LA PAROLE

Etat de l'art

1. Base de données
2. Présentation modèles sans deep learning
3. Présentation modèles avec deep learning

Présentation des bases de données

Base de donnée

EMO-DB

(emotional database berlin)

CREMA D

(Crowd Sourced Emotional Multimodal Actors Dataset)

RAVDESS

IMEOCAP

Présentation des bases de données

Base de donnée	Taille
EMO-DB (emotional database berlin)	40 MB
CREMA D (Crowd Sourced Emotional Multimodal Actors Dataset)	473 MB
RAVDESS	24.8 GB
IMEOCAP	?

Présentation des bases de données

Base de donnée	Taille	Autres
EMO-DB (emotional database berlin)	40 MB	-En Allemand -10 Orateurs : 5h, 5f -freq echantillonage : 16kHz
CREMA D (Crowd Sourced Emotional Multimodal Actors Dataset)	473 MB	-91 orateurs -12 phrases différentes
RAVDESS	24.8 GB	-24 orateurs -fréquence d'échantillonage 48kHz
IMEOCAP	?	Pas d'accès public

Présentation des bases de données

Base de donnée	Taille	Autres	Retenue
EMO-DB (emotional database berlin)	40 MB	-En Allemand -10 Orateurs : 5h, 5f -freq echantillonage : 16kHz	OUI
CREMA D (Crowd Sourced Emotional Multimodal Actors Dataset)	473 MB	-91 orateurs -12 phrases différentes	OUI
RAVDESS	24.8 GB	-24 orateurs -fréquence d'échantillonage 48kHz	NON
IMEOCAP	?	Pas d'accès public	NON

TRAITEMENT DE LA PAROLE

Etat de l'art

1. Base de données

2. Présentation modèles sans deep learning

Comparaison des différents modèles d'apprentissage selon l'état de l'art

Modèle	Inconvénients	Avantages	Performance
KNN (K nearest neighbours)	Classificateur Linéaire pour un problème non linéaire !	Facile à implémenter	Très mauvaise
Modèle de mélange gaussien	<ul style="list-style-type: none"> Difficulté Théorique Difficulté d'implémentation 	x	Bonne
Forêt aléatoire	x	x	Très bonne
Machines à vecteurs de support	x	Facile à implémenter	Très bonne
Modèles de Markov cachés	Apprentissage difficile	x	Moyenne

Comparaison des modèles classiques sur les bases de données utilisées

Modèle\BDD	EMO-DB	CREMA D
SVM		
Random Forest		

Source : [1],[2],[3],[4],[5],[6],[7],[8]

Comparaison des modèles classiques sur les bases de données utilisées

Modèle\BDD	EMO-DB	CREMA D
SVM	83.43 (min 77, max 93, N = 8)	
Random Forest	84.76 (min 80, max 89, N = 5)	

Source : [1],[2],[3],[4],[5],[6],[7],[8]

Comparaison des modèles classiques sur les bases de données utilisées

Modèle\BDD	EMO-DB	CREMA D
SVM	83.43 (min 77, max 93, N = 8)	60.43 (min 56, max 63, N = 4)
Random Forest	84.76 (min 80, max 89, N = 5)	65.00 (min = 60, max = 70, N =2)

Source : [1],[2],[3],[4],[5],[6],[7],[8]

Comparaison des modèles classiques sur les bases de données utilisées

Modèle\BDD	EMO-DB	CREMA D
SVM	83.43 (min 77, max 93, N = 8)	60.43 (min 56, max 63, N = 4)
Random Forest	84.76 (min 80, max 89, N = 5)	65.00 (min = 60, max = 70, N = 2)
HUMAIN	80,9	70

Source : [1],[2],[3],[4],[5],[6],[7],[8]

Comparaisons des modèles de Deep Learning

Modèle	Inconvénients	Avantages	Performance
MLP	Pas adapté aux signaux temporels	Facile à implémenter	Moyenne
CNN	Pas adapté aux signaux temporels	Repère les motifs	Bonne
LSTM	Temps de calcul	Modélise le relations temporelles	Très bonne
TRANSFORMER	Difficulté théorique	Performances	Très bonne

TRAITEMENT DE LA PAROLE

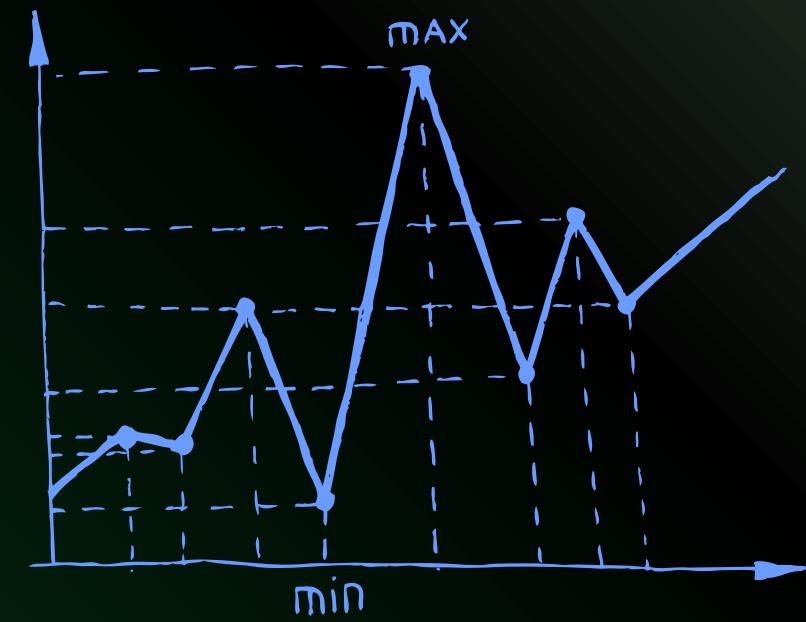
Préparation des données à l'apprentissage.

Objectif

Extraire les meilleures données possibles pour la phase d'apprentissage.

Pourquoi ne pas garder le signal brut?

Trop complexe, présence de bruit.



Quelles caractéristiques de la voix sont utiles à la détection d'émotions?

De nombreuses caractéristiques disponibles

Energie

Sonic
Energie moyenne
Zéro Crossing Rate

Formants

Amplitudes spectrales sur des bandes de fréquences où l'énergie acoustique est particulièrement intense dans le spectre sonore

Pitch & Jitter

Fréquence fondamentale & Ses Variations court terme

$$R(\tau) = \sum_{n=0}^{N-1} x(n) \cdot x(n + \tau)$$

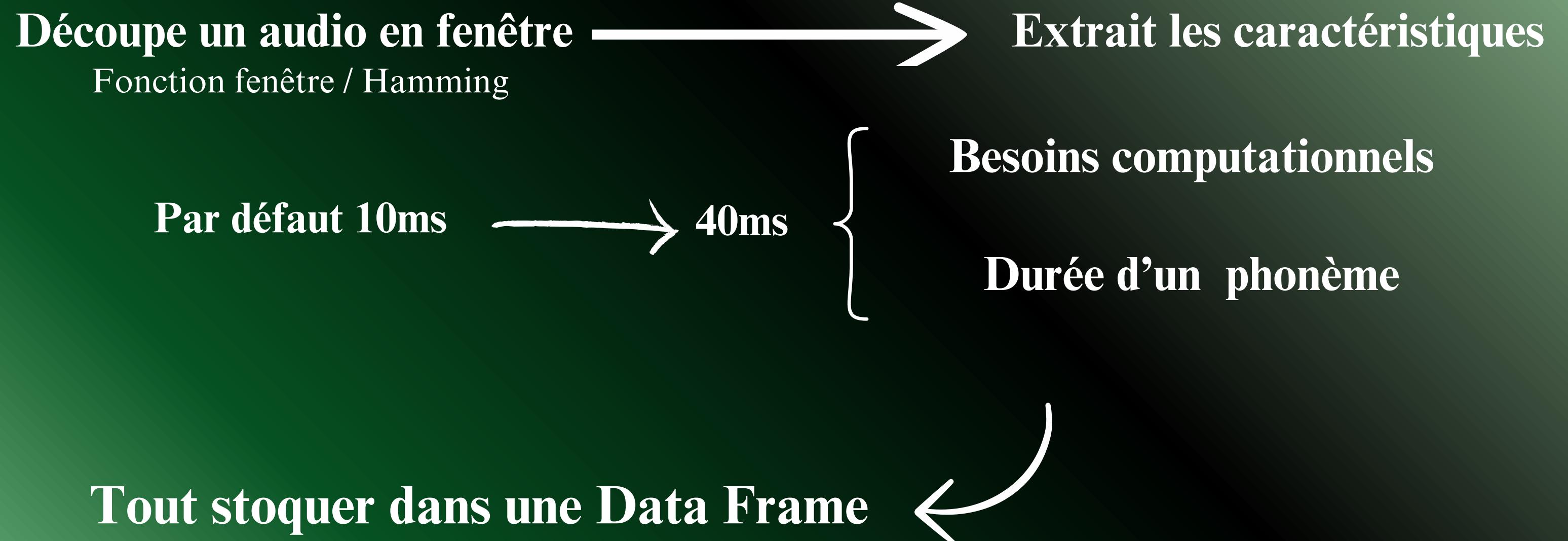
MFCC

Transformée de Fourier
Echelle de Mel
Transformation cepstrale

$$\text{MFCC}_n = \sum_{m=1}^M \log(S_m) \cdot \cos \left[n \cdot \frac{\pi(m - 0.5)}{M} \right]$$

Première Solution: OpenSMILE

Etape 1 : Extraction des données



Première Solution: OpenSMILE

Etape 2 : Normalisation

Chaque caractéristique → Normalisée entre 0 et 1

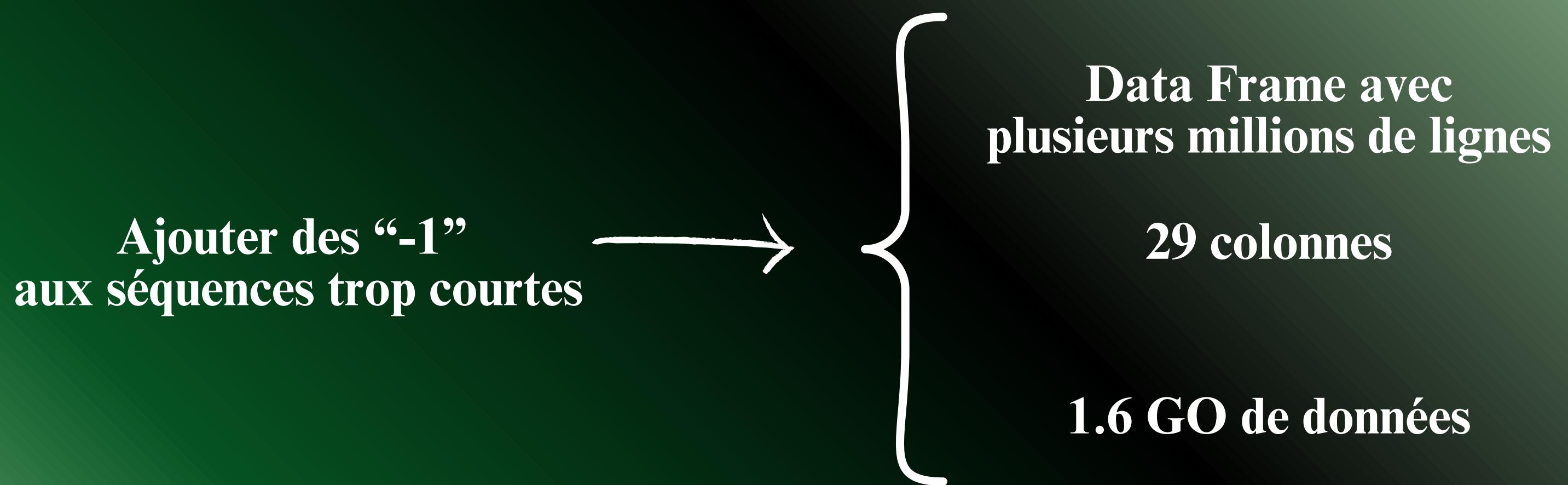


En séparant les bases de données

Quel Intérêt? → Généralisation des données

Première Solution: OpenSMILE

Etape 3 : Remplissage



Optimisation du choix des Features avec l'Information Mutuelle

Qu'est ce que l'information mutuelle? (MI)

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)}$$

Evalue la proportion dans laquelle la connaissance d'une variable réduit l'incertitude sur une autre.

Défis :

- Lien MI/prédicibilité des labels ?
- Redondance entre features

Processus suivi :

Extraction des features

MFCC / Jitter / Shimmer



Calcul du MI



Validation avec
modèles

SVM / Régression Logistique



Sélection des features
les plus pertinentes

TRAITEMENT DE LA PAROLE

Présentation détaillée des modèles classiques

- 1. Random Forest**
- 2. Régression Logistique**
- 3. SVM**

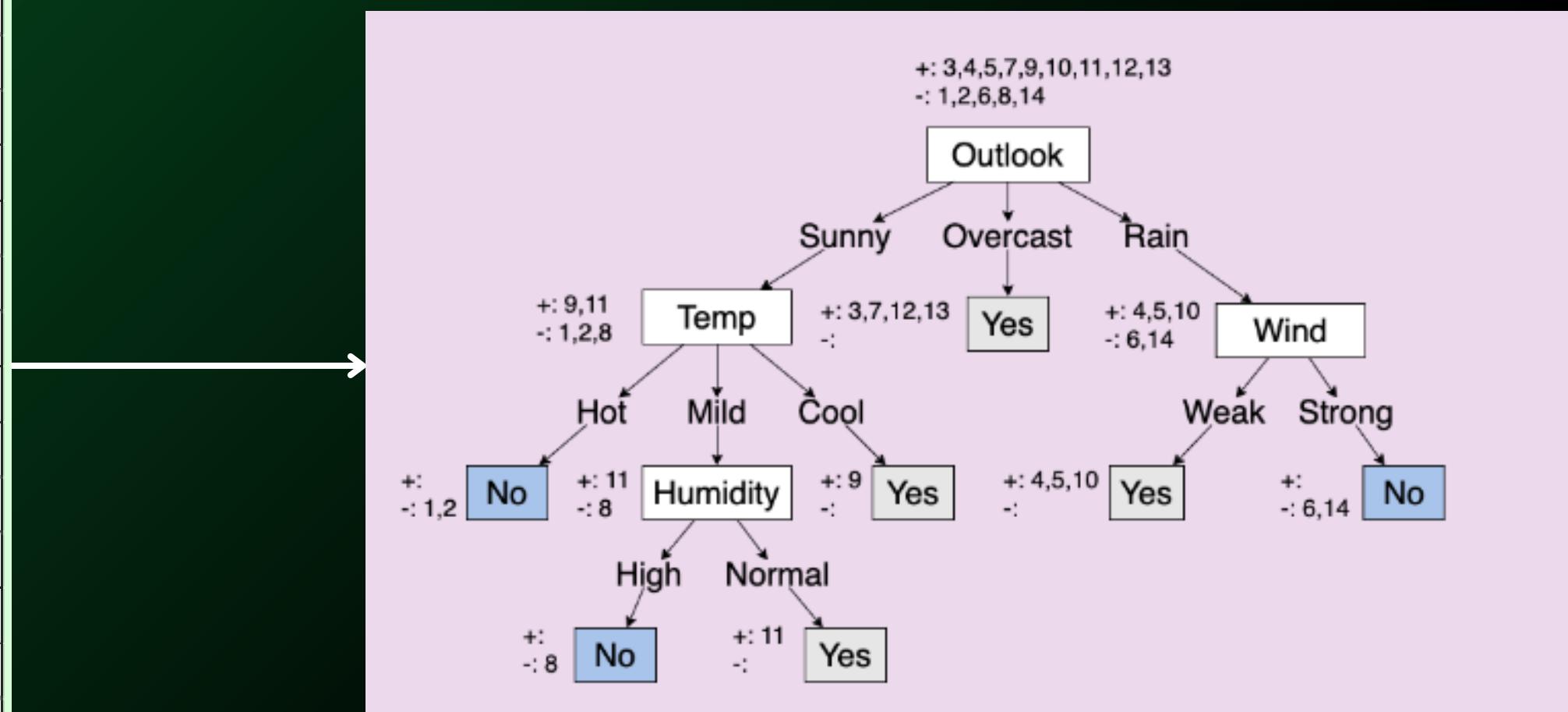
1. RANDOM FOREST

RANDOM FOREST: avantages/incovénients

AVANTAGES	INCONVENIENTS
<ul style="list-style-type: none">-Permet de résoudre les problèmes de classification supervisée avec plusieurs catégories-Marche bien avec de gros Datasets-Peut modéliser relations non-linéaires (cf exemple)-Marchait bien dans la bibliographie	<ul style="list-style-type: none">-non adapté aux données temporelles

Arbres de décisions

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No
15	Sunny	Hot	High	Weak	No



Arbres de décisions

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Définition de l'entropie du dataset :

$$I(P(c_1), \dots, P(c_k)) = - \sum_{i=1}^k P(c_i) \log_2(P(c_i)).$$

Arbres de décisions

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Définition de l'entropie du dataset :

$$I(P(c_1), \dots, P(c_k)) = - \sum_{i=1}^k P(c_i) \log_2(P(c_i)).$$

Entropie de ce dataset

$$I\left(\frac{9}{14}, \frac{5}{14}\right) = -\frac{9}{14} * \log_2\left(\frac{9}{14}\right) - \frac{5}{14} * \log_2\left(\frac{5}{14}\right) = 0.940.$$

Arbres de décisions

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Gain d'information apporté par la feature A

$$Gain(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - \sum_{i=1}^k \frac{p_i + n_i}{p+n} * I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

Arbres de décisions

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Gain d'information apporté par la feature A

$$Gain(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - \sum_{i=1}^k \frac{p_i + n_i}{p+n} * I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

Gain d'info apporté par Outlook

Gain(Outlook)

$$\begin{aligned} &= 0.940 - \left(\frac{5}{14} I\left(\frac{2}{5}, \frac{3}{5}\right) + \frac{4}{14} I\left(\frac{4}{4}, \frac{0}{4}\right) + \frac{5}{14} I\left(\frac{3}{5}, \frac{2}{5}\right) \right) \\ &= 0.940 - \left(\frac{5}{14} 0.971 + \frac{4}{14} 0 + \frac{5}{14} 0.971 \right) \end{aligned}$$

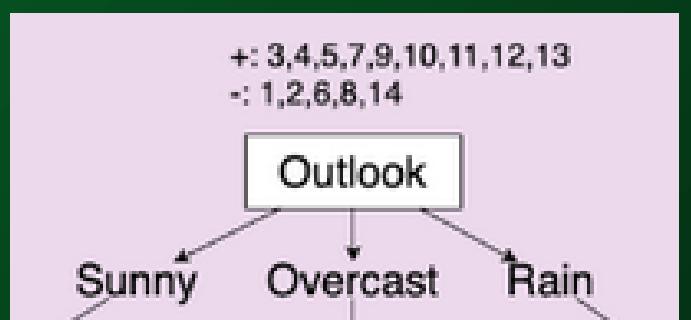
$$= 0.940 - 0.694$$

$$= 0.247.$$

Arbres de décisions

Algorithm 1 ID3 Algorithm (Features, Examples)

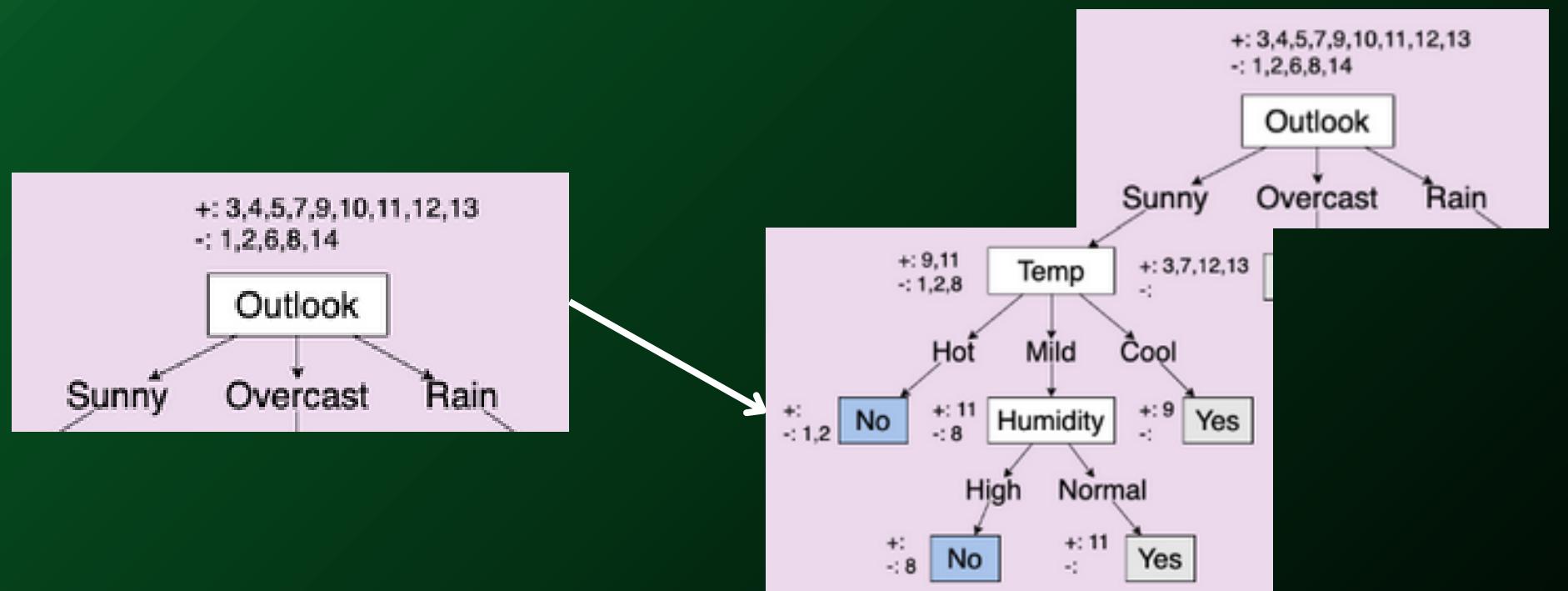
```
1: If all examples are positive, return a leaf node with decision yes.  
2: If all examples are negative, return a leaf node with decision no.  
3: If no features left, return a leaf node with the most common decision of the examples.  
4: If no examples left, return a leaf node with the most common decision of the examples in the parent.  
5: else  
6:   choose the most important feature  $f$   
7:   for each value  $v$  of feature  $f$  do  
8:     add arc with label  $v$   
9:     add subtree  $ID3(F - f, s \in S | f(s) = v)$   
10:  end for
```



Arbres de décisions

Algorithm 1 ID3 Algorithm (Features, Examples)

```
1: If all examples are positive, return a leaf node with decision yes.  
2: If all examples are negative, return a leaf node with decision no.  
3: If no features left, return a leaf node with the most common decision of the examples.  
4: If no examples left, return a leaf node with the most common decision of the examples in the parent.  
5: else  
6:   choose the most important feature  $f$   
7:   for each value  $v$  of feature  $f$  do  
8:     add arc with label  $v$   
9:     add subtree  $ID3(F - f, s \in S | f(s) = v)$   
10:  end for
```



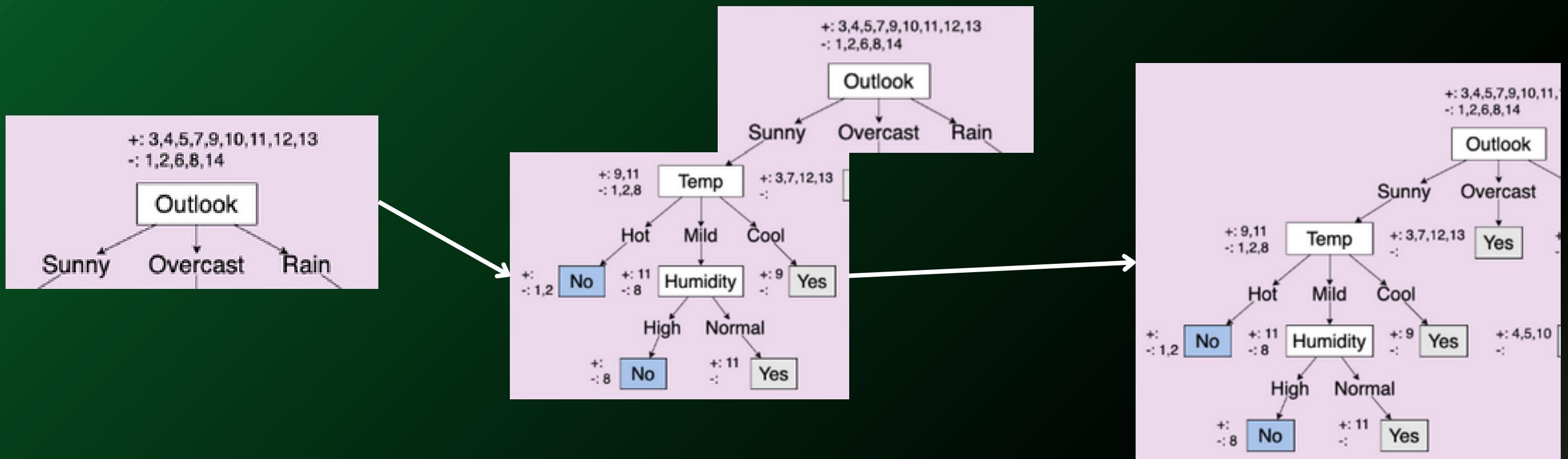
Arbres de décisions

Algorithm 1 ID3 Algorithm (Features, Examples)

```

1: If all examples are positive, return a leaf node with decision yes.
2: If all examples are negative, return a leaf node with decision no.
3: If no features left, return a leaf node with the most common decision of the examples.
4: If no examples left, return a leaf node with the most common decision of the examples in the parent.
5: else
6:   choose the most important feature  $f$ 
7:   for each value  $v$  of feature  $f$  do
8:     add arc with label  $v$ 
9:     add subtree  $ID3(F - f, s \in S | f(s) = v)$ 
10:  end for

```



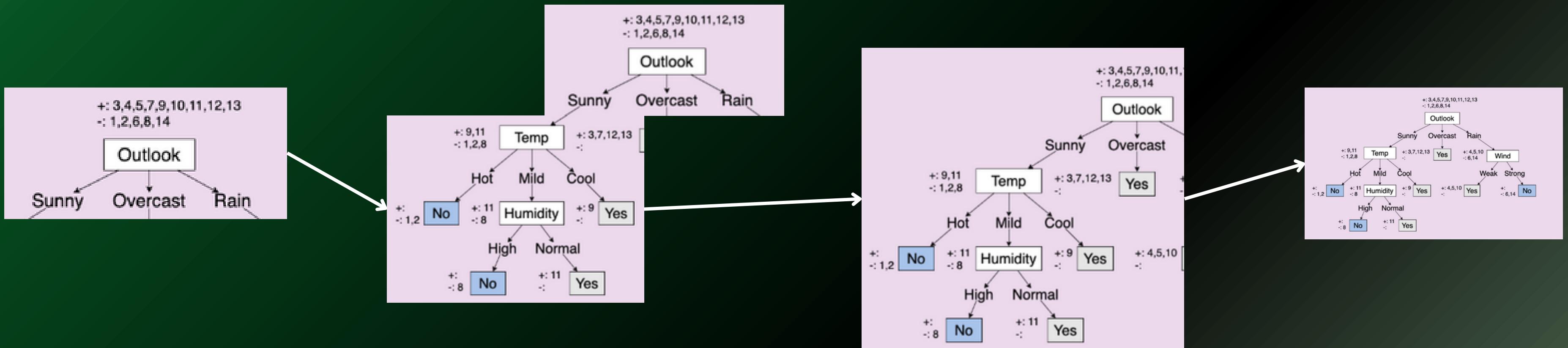
Arbres de décisions

Algorithm 1 ID3 Algorithm (Features, Examples)

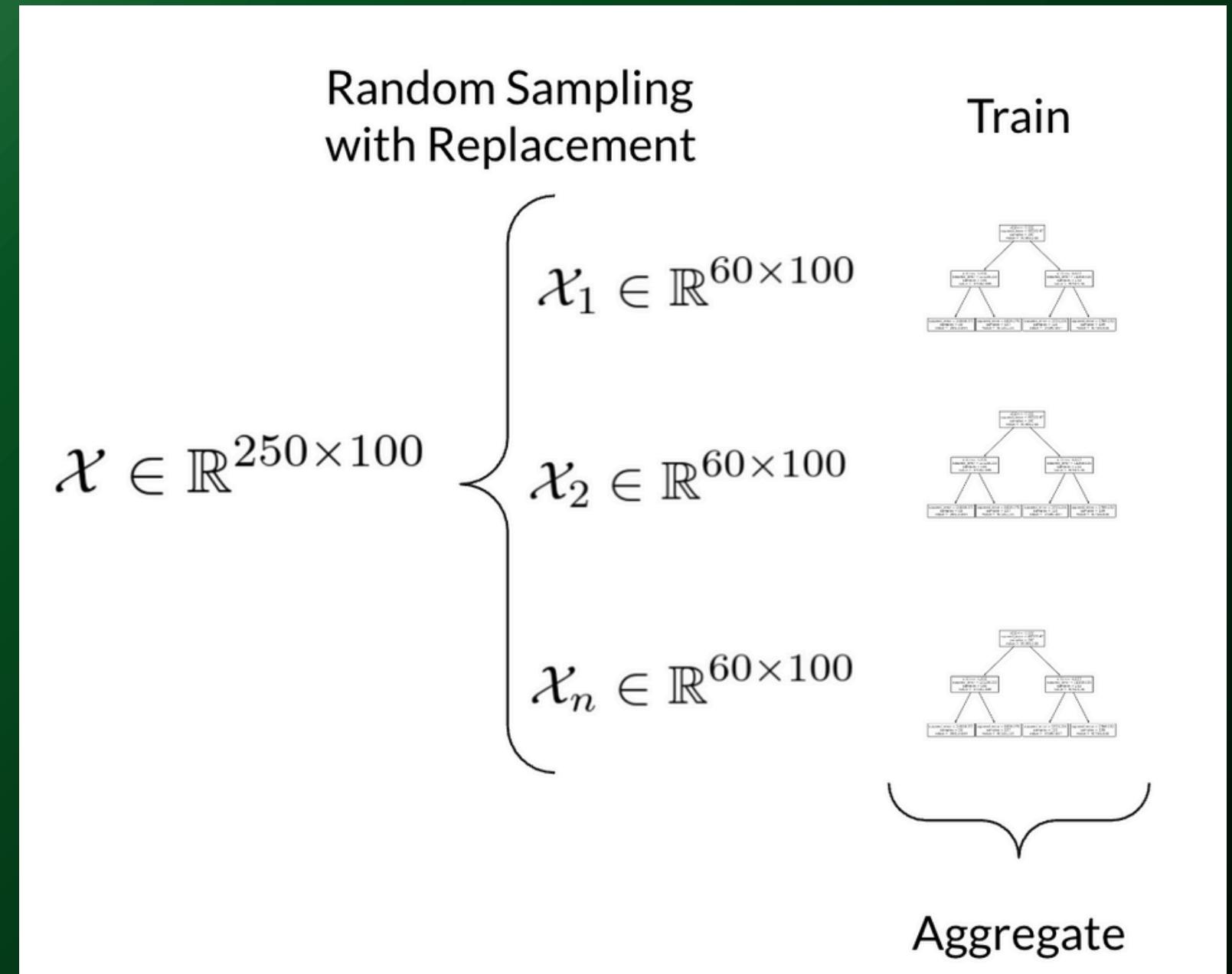
```

1: If all examples are positive, return a leaf node with decision yes.
2: If all examples are negative, return a leaf node with decision no.
3: If no features left, return a leaf node with the most common decision of the examples.
4: If no examples left, return a leaf node with the most common decision of the examples in the parent.
5: else
6:   choose the most important feature  $f$ 
7:   for each value  $v$  of feature  $f$  do
8:     add arc with label  $v$ 
9:     add subtree  $ID3(F - f, s \in S | f(s) = v)$ 
10:  end for

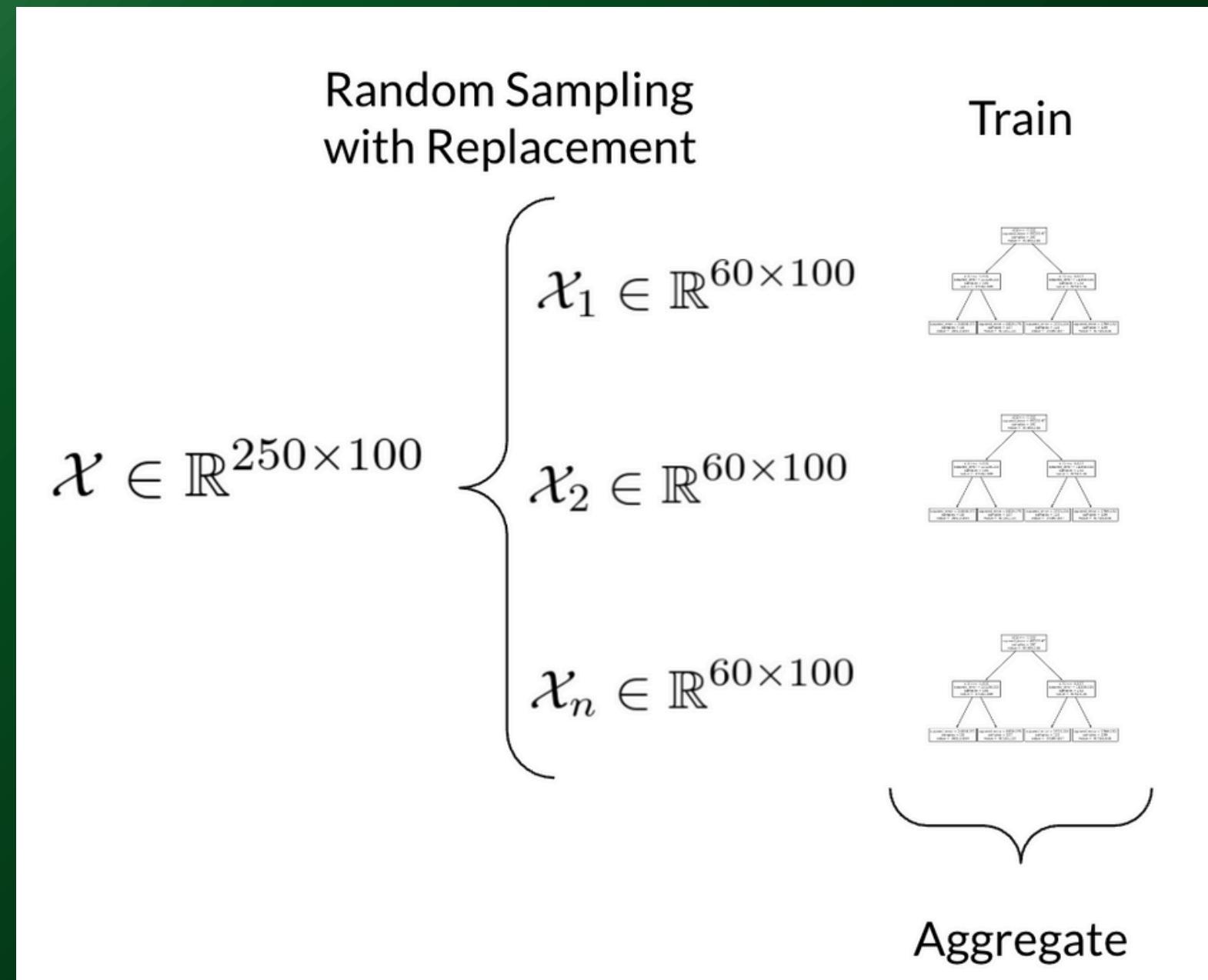
```



Forêts aléatoires



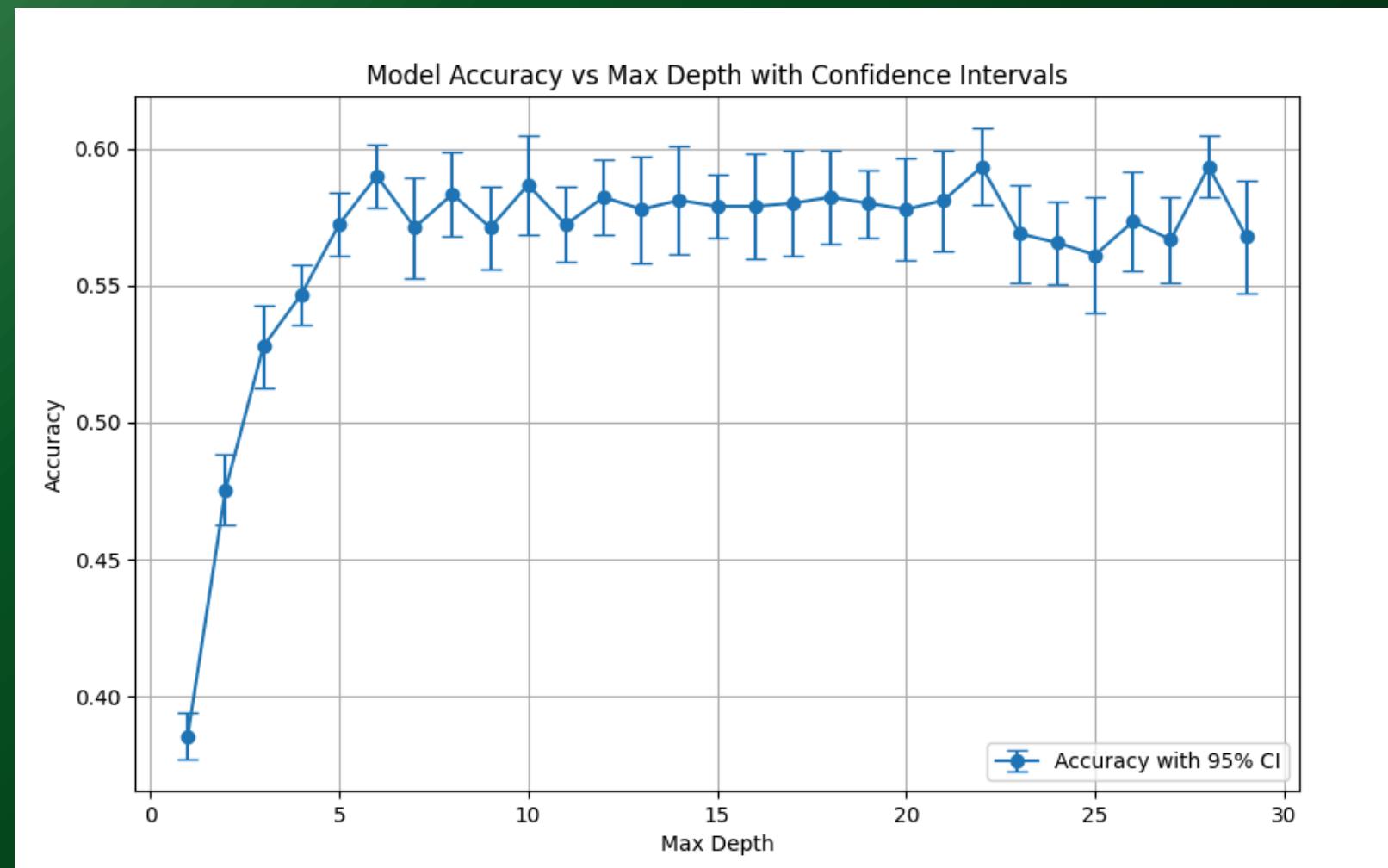
Forets aléatoires



$$C^*(x) = \underset{y \in Y}{\operatorname{argmax}} \# \{i : C_i(x) = y\}$$

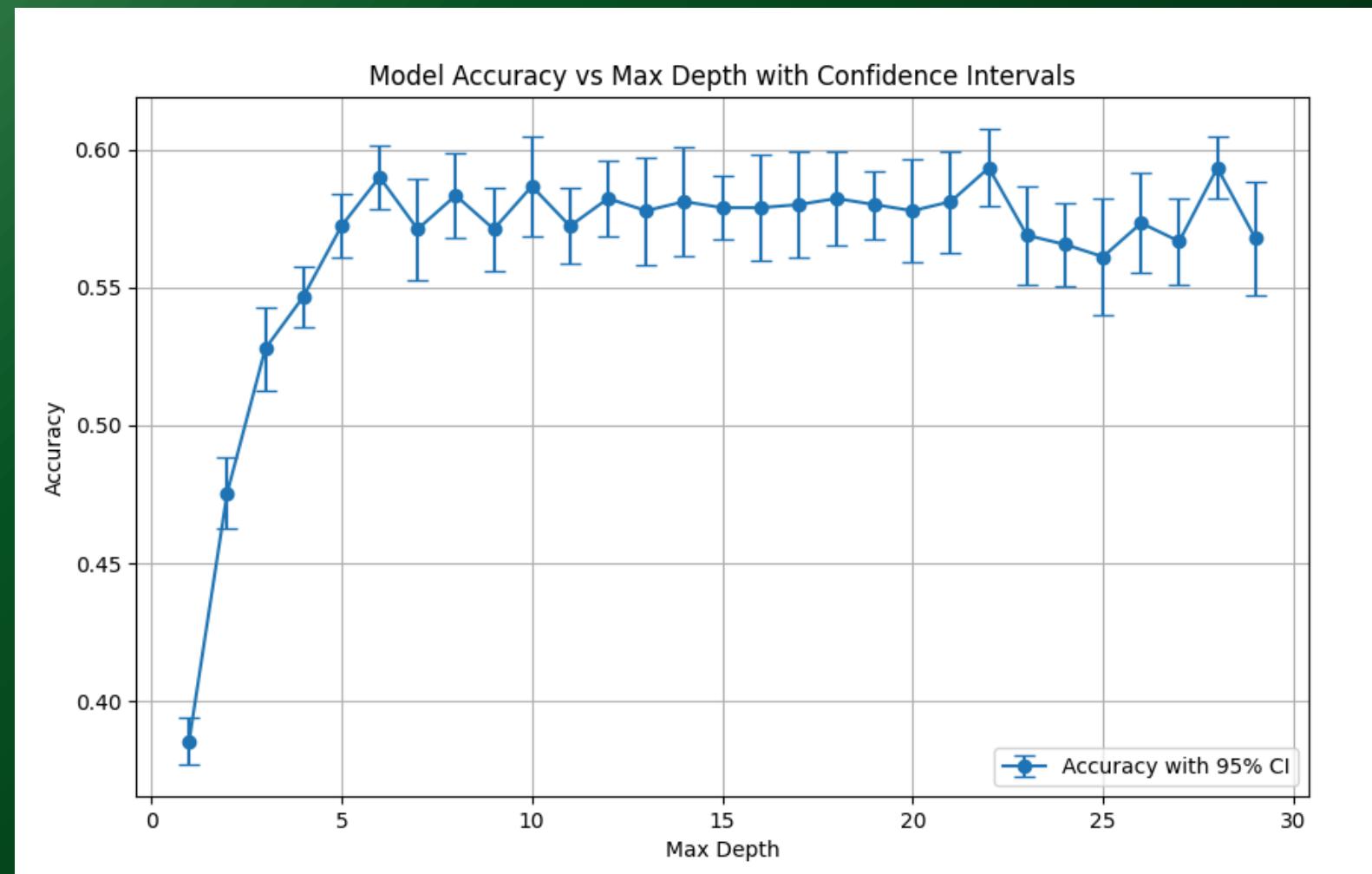
Forêts aléatoires : optimisation des paramètres/features

Librairie opensmile, EmoDB

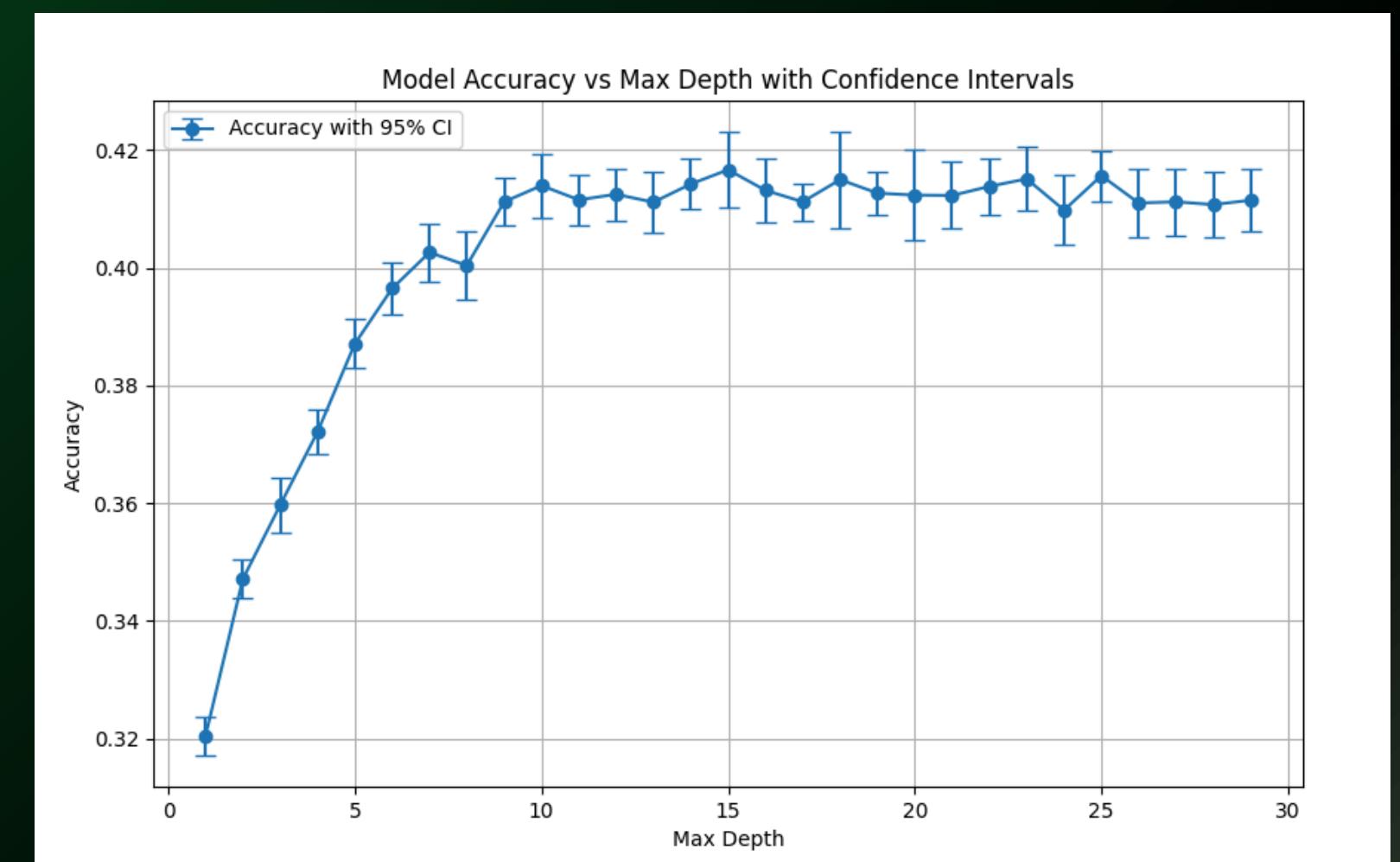


Forêts aléatoires : optimisation des paramètres/features

Librairie opensmile, EmoDB

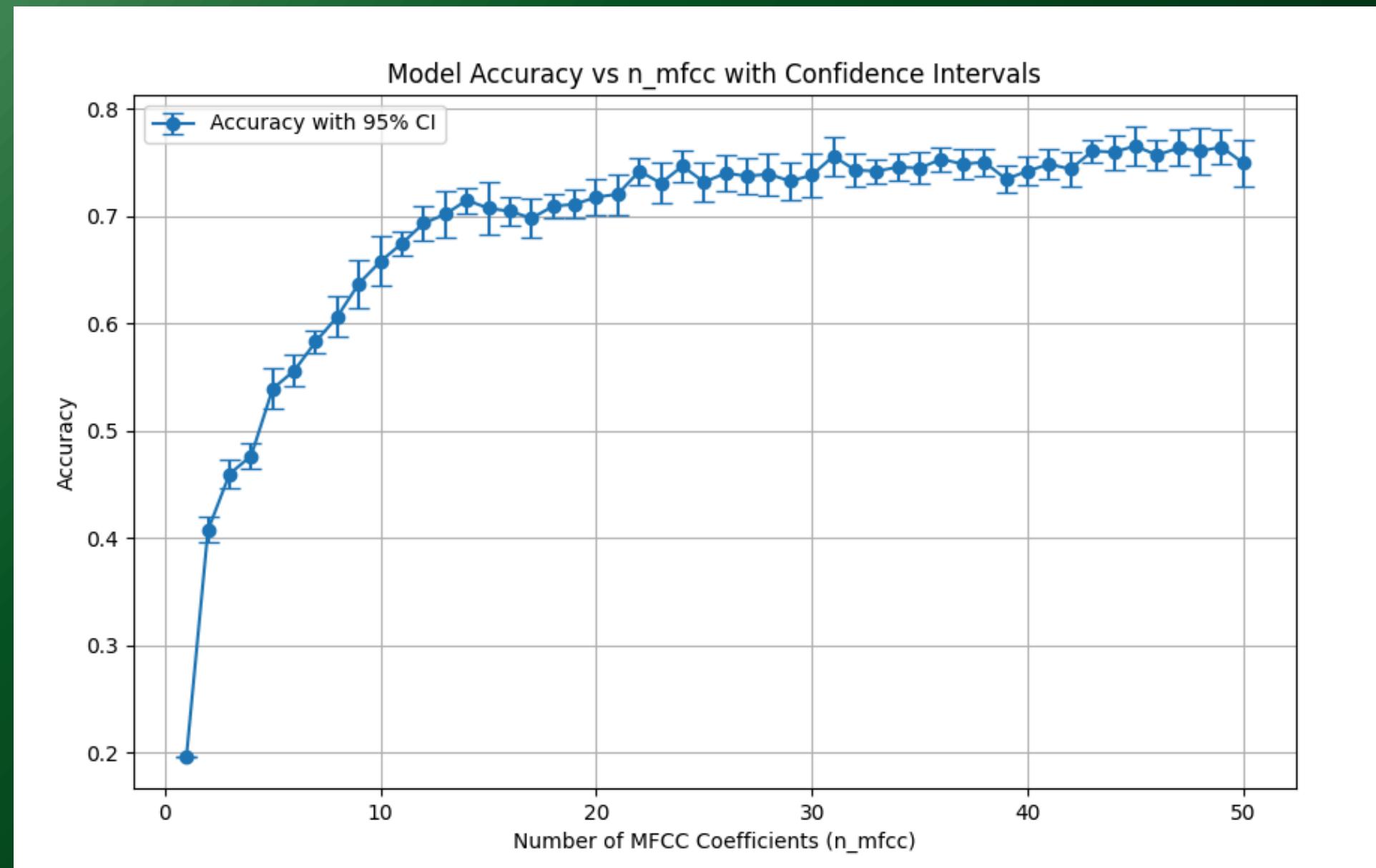


Librairie opensmile, Crema D



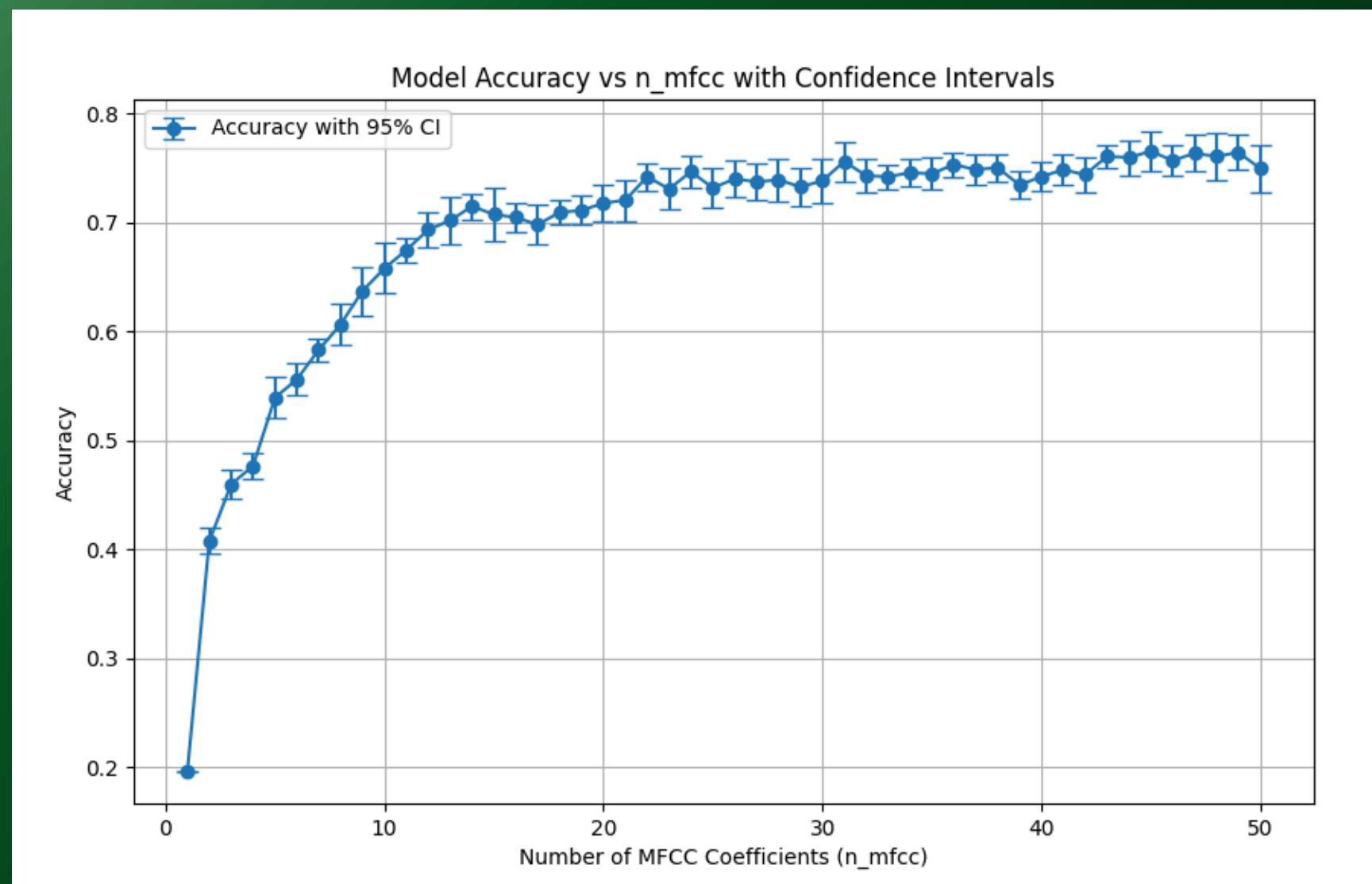
Forêts aléatoires : optimisation des paramètres/features

Librairie librosa, EmoDB

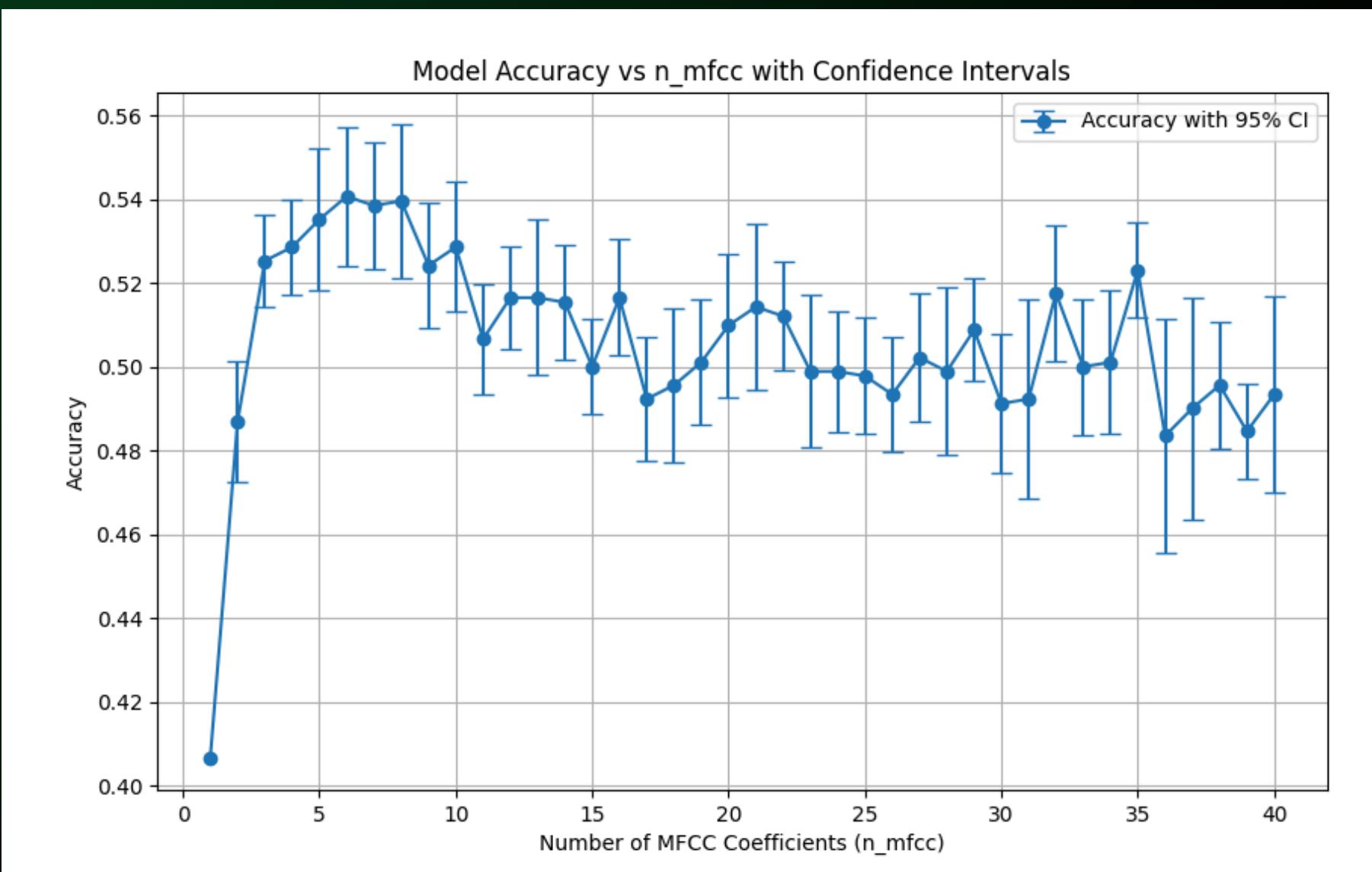


Forêts aléatoires : optimisation des paramètres/features

Librairie librosa, EmoDB

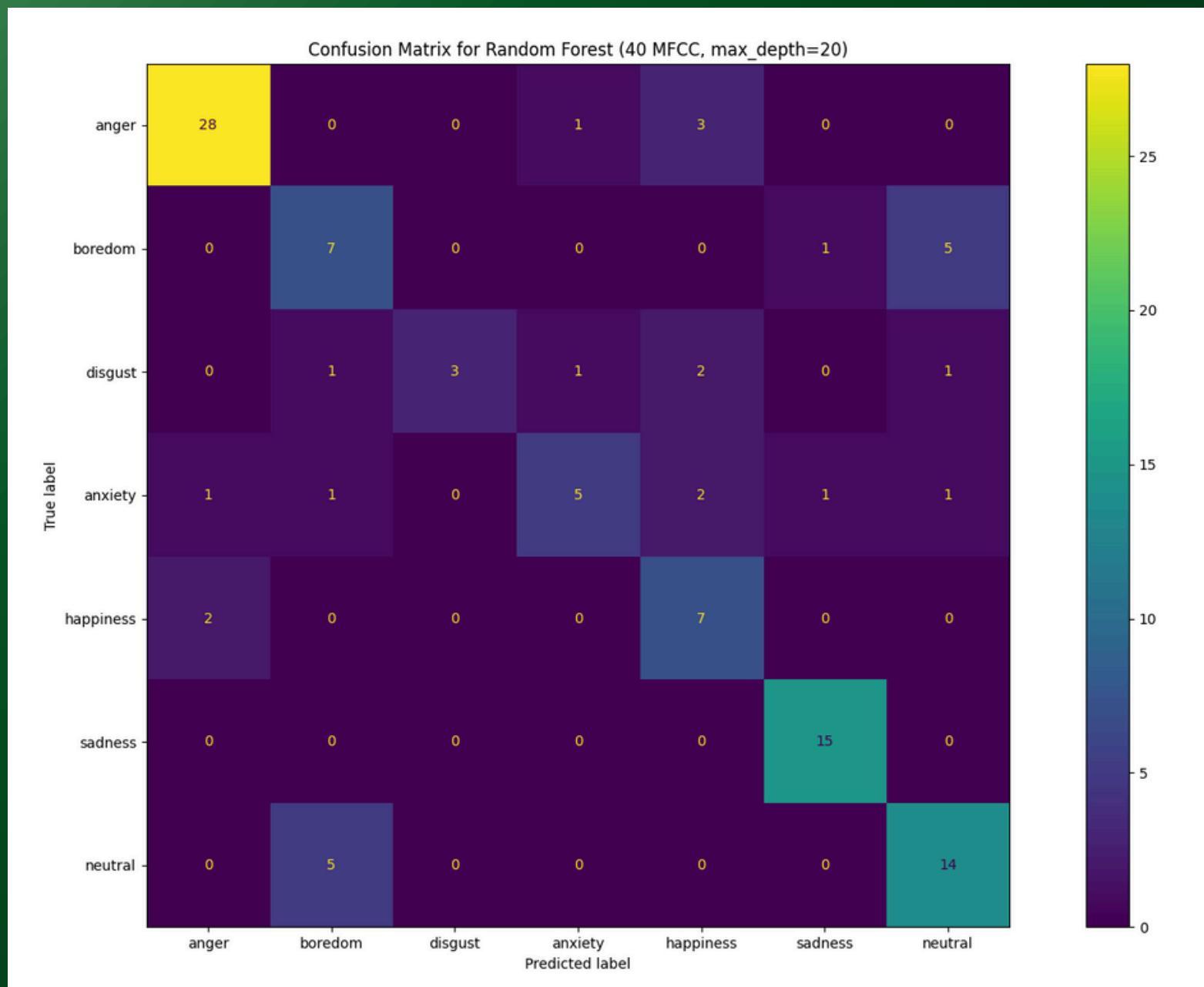


Librairie librosa, CremaD

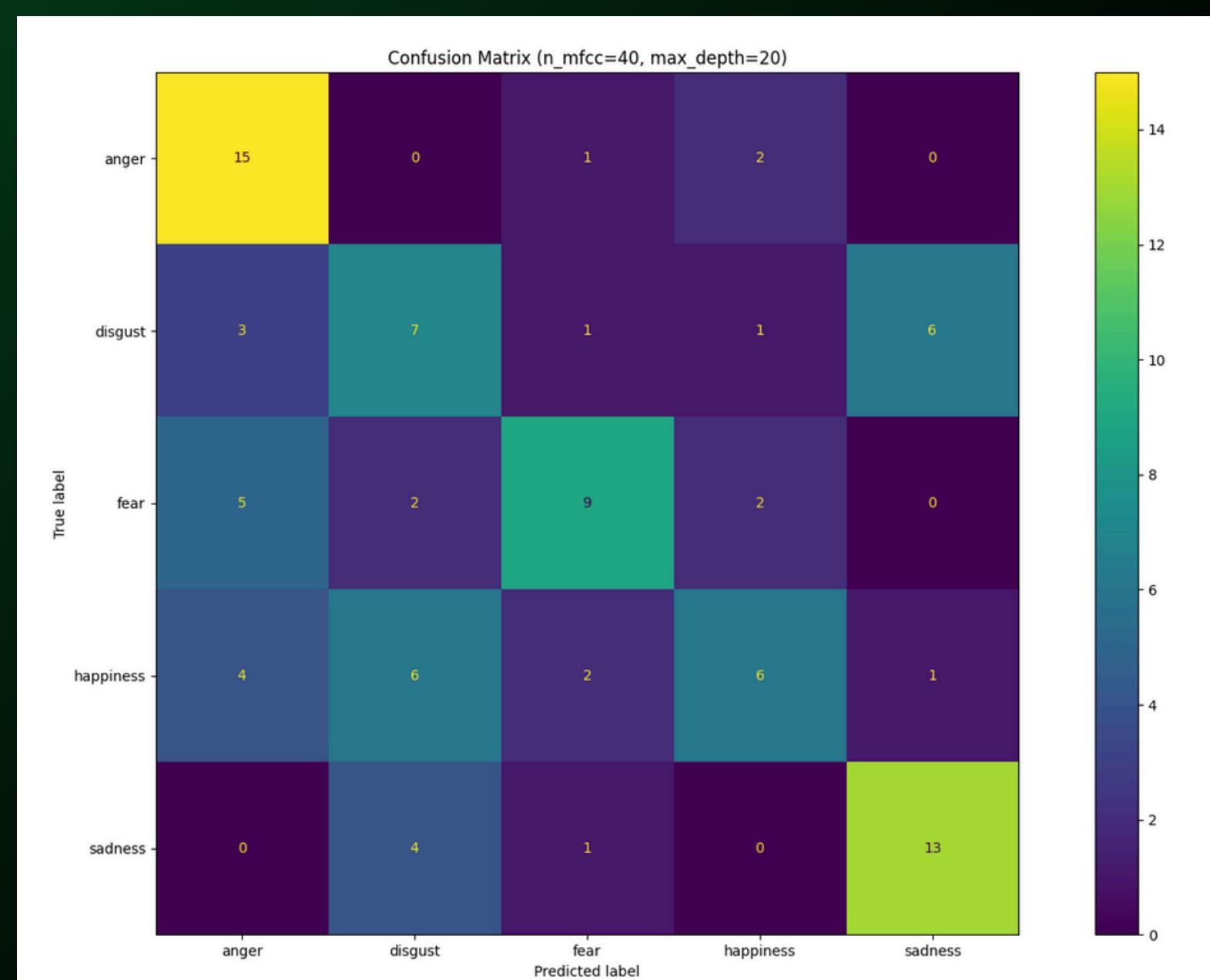


Forêts aléatoires : Résultats finaux

Librairie librosa, EmoDB



Librairie librosa, Crema D



2. RÉGRESSION LOGISTIQUE

Qu'est ce qu'une régression logistique ?

Algorithme d'IA utilisé pour la classification multi-classes.

Fonctionnement :

Transforme l'espace des données en un gradient de probabilité différent pour chaque classe

Attribue des coefficients β aux features pour générer une valeur z :

$$z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

Puis convertit cette valeur en une probabilité entre 0 et 1 à l'aide d'une sigmoïde.

Avantages :

- Entraînement rapide
- Interprétabilité
- Gestion de l'incertitude par les probabilités

Défauts :

- Ne capture pas les relations non linéaires
- très sensible au bruit

Résultats

Test sur Emo-DB

Features :

13 MFCC

Precision finale :

68%

Classification Report:					
	precision	recall	f1-score	support	
anger	0.74	0.88	0.81	26	
anxiety	0.64	0.50	0.56	14	
boredom	0.50	0.31	0.38	16	
disgust	0.67	0.67	0.67	9	
happiness	0.69	0.64	0.67	14	
neutral	0.63	0.75	0.69	16	
sadness	0.79	0.92	0.85	12	
accuracy			0.68	107	
macro avg	0.66	0.67	0.66	107	
weighted avg	0.67	0.68	0.67	107	

3. MACHINE A VECTEUR DE SUPPORT (SVM)

Fonctionnement :

Fait pour séparer deux classes :

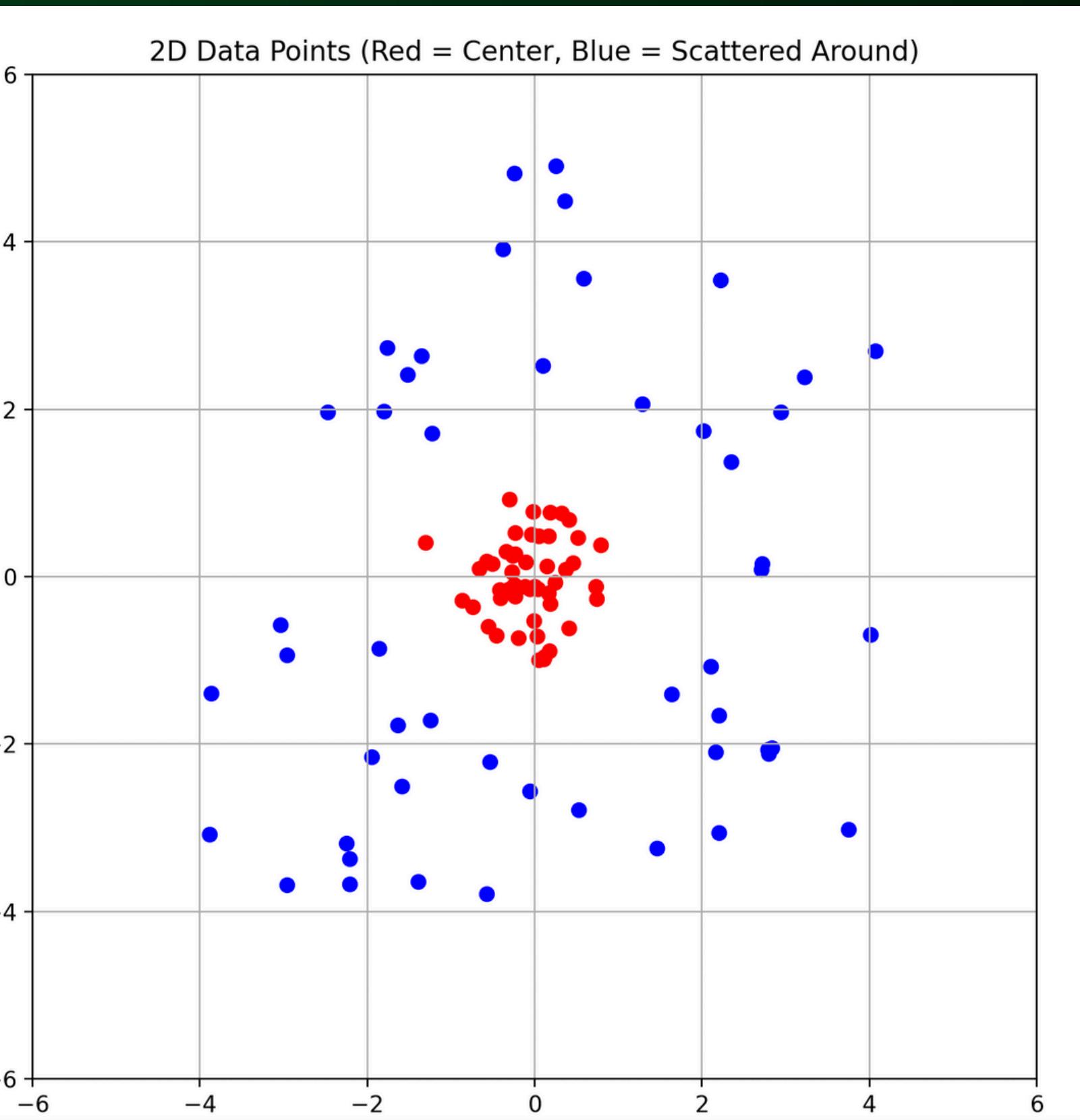
Recherche l'Hyperplan qui maximise la 'marge'

le SVM utilise une transformation mathématique (fonction noyau) pour projeter les données dans un espace de dimension supérieure où elles deviennent séparables.

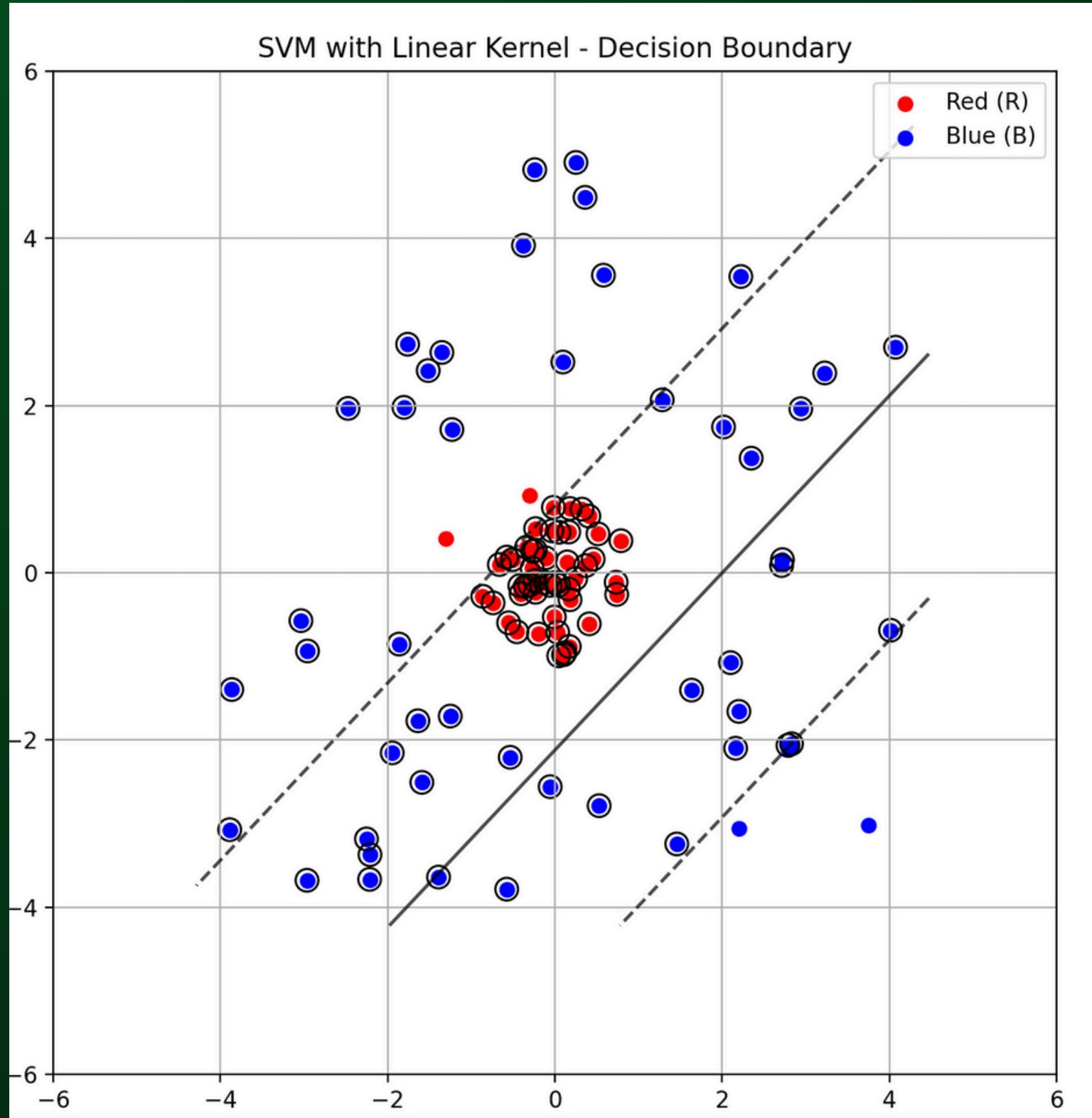
Exemple :

Deux catégories :
Blue / Red

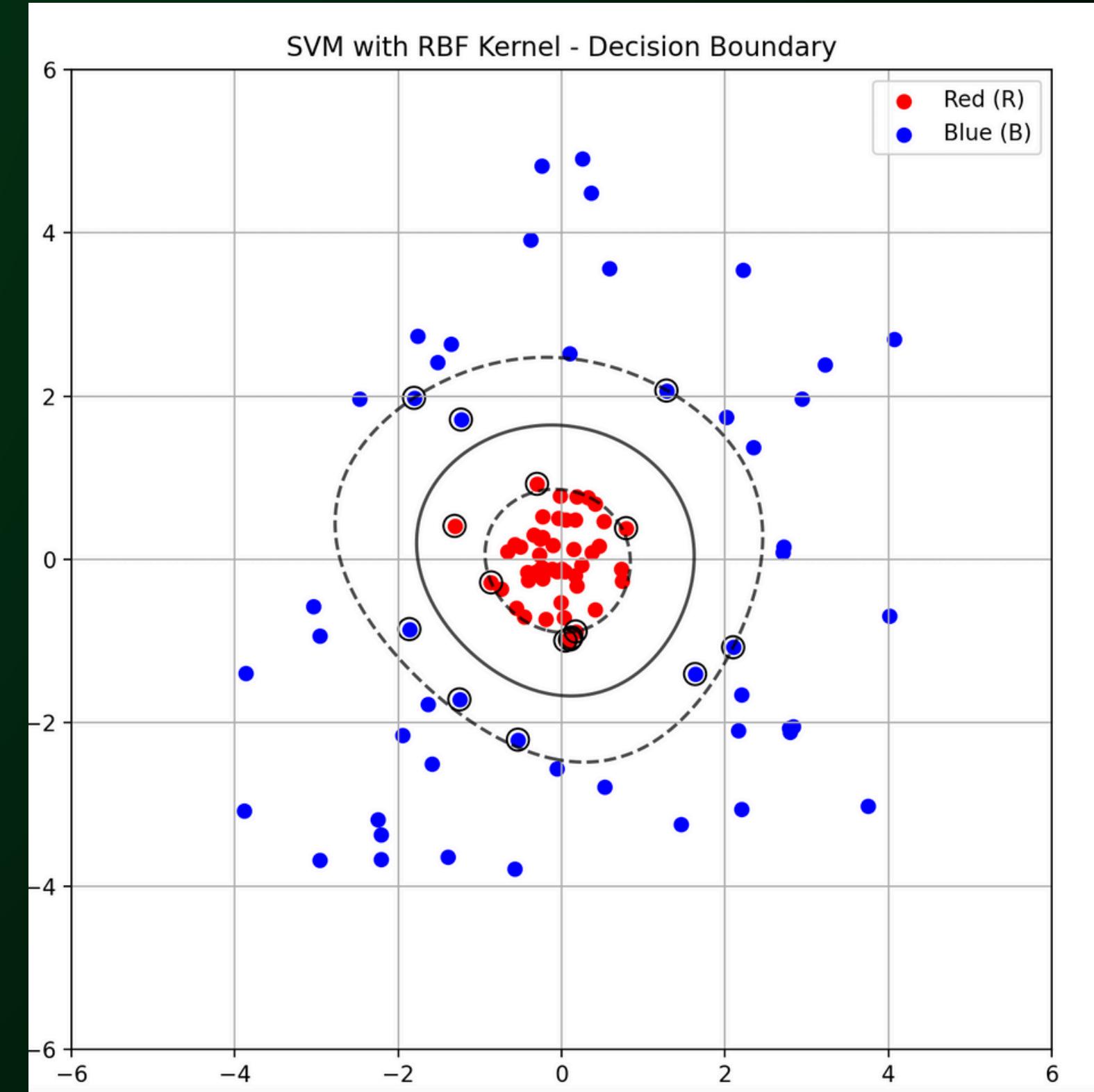
Données non linéairement
séparables



Linear Kernel



RBF Kernel



Classification multi-classes?

Comparaison **one to one** :

Chaque SVM compare deux classes à la fois.

Si n est le nombre de classes, il faut entraîner $n(n-1)/2$ SVM.

Exemple avec 3 classes :

SVM1 : Neutre vs Joyeux.

SVM2 : Neutre vs Triste.

SVM3 : Joyeux vs Triste.

Prise de décision par vote majoritaire

Première Implémentation :

Test sur EMO-DB

features :

13 MFCCs

Précision initiale :

71%

Classification Report:

	precision	recall	f1-score	support
anger	0.76	0.85	0.80	26
anxiety	0.58	0.50	0.54	14
boredom	0.75	0.38	0.50	16
disgust	0.75	0.67	0.71	9
happiness	0.64	0.64	0.64	14
neutral	0.68	0.94	0.79	16
sadness	0.79	0.92	0.85	12
accuracy			0.71	107
macro avg	0.71	0.70	0.69	107
weighted avg	0.71	0.71	0.70	107

Phase d'optimisation :

Première étape :
optimisation des hyperparamètres

Hyperparamètres en question :
C : degré de régularisation
Kernel
Degré (pour Kernel polynomial)

Méthode :
Grid-Search avec
C : [0.1:100]
Kernel : rbf, polynomial, linéaire
Degré : [1:7]

Résultats de la Grid-Search

Meilleur modèle ?

- précision
- F1-score
- Cross-Validation

Résultats :

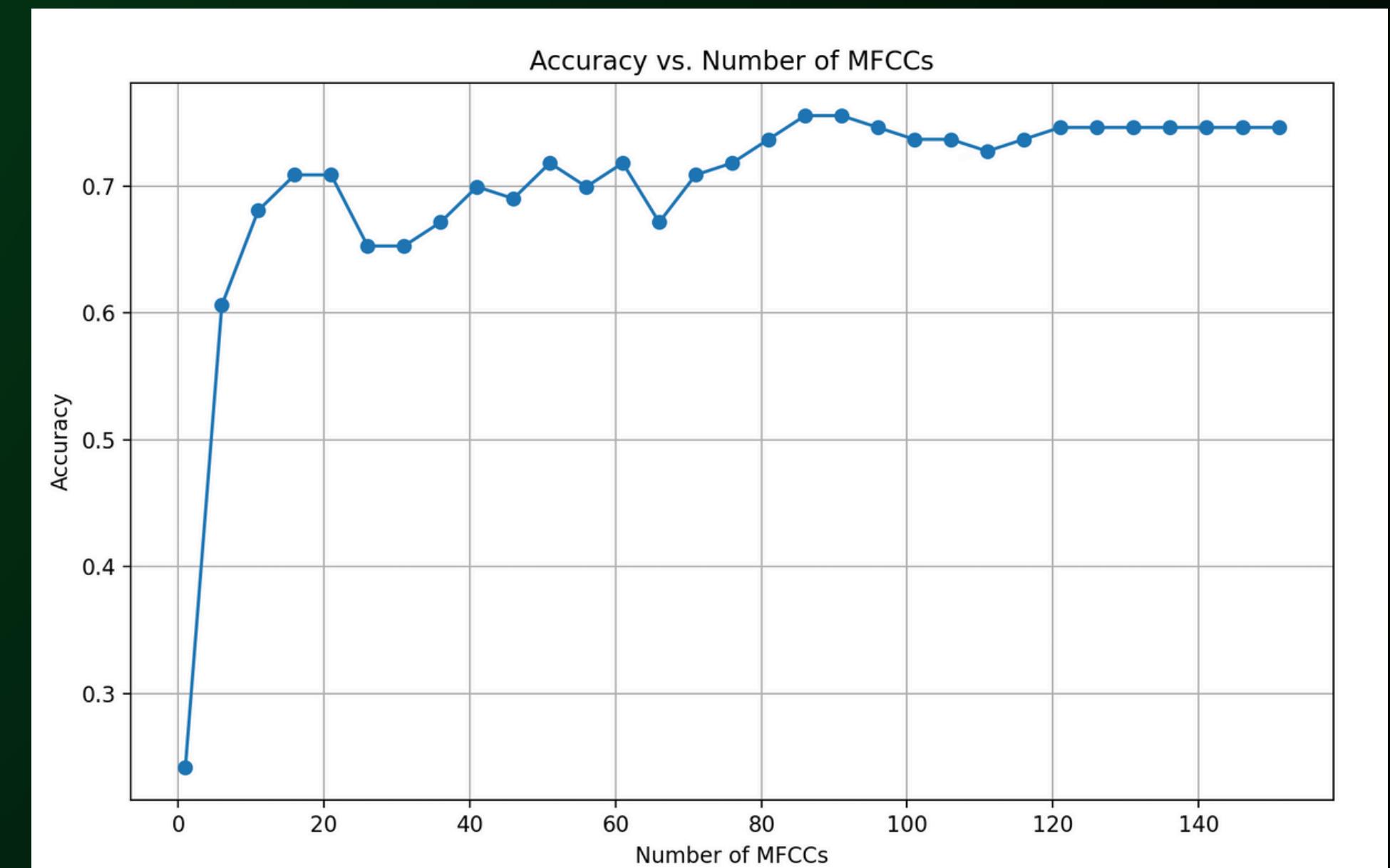
- $C = 0.1$
- Kernel = linear

Suite : Optimisation des features

Optimisation naïve des MFCCs

Première recherche :
quel nombre de MFCC
passer en entrée?

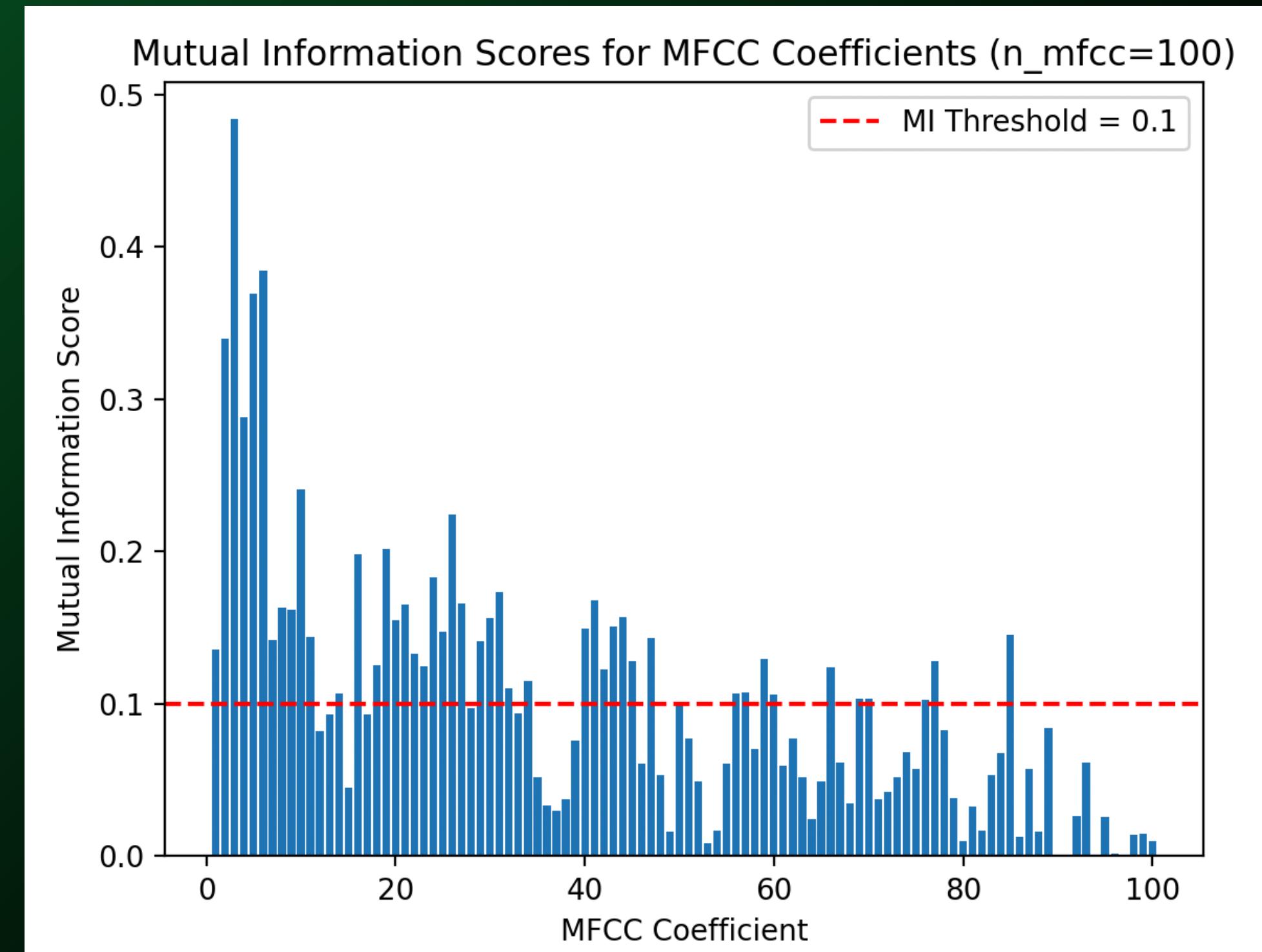
$n_{max} = 86$
precision = 75,4%



Optimisation des features avec MI

MI entre MFCC[i] et labels

MFCC utile ->
score MI > threshold

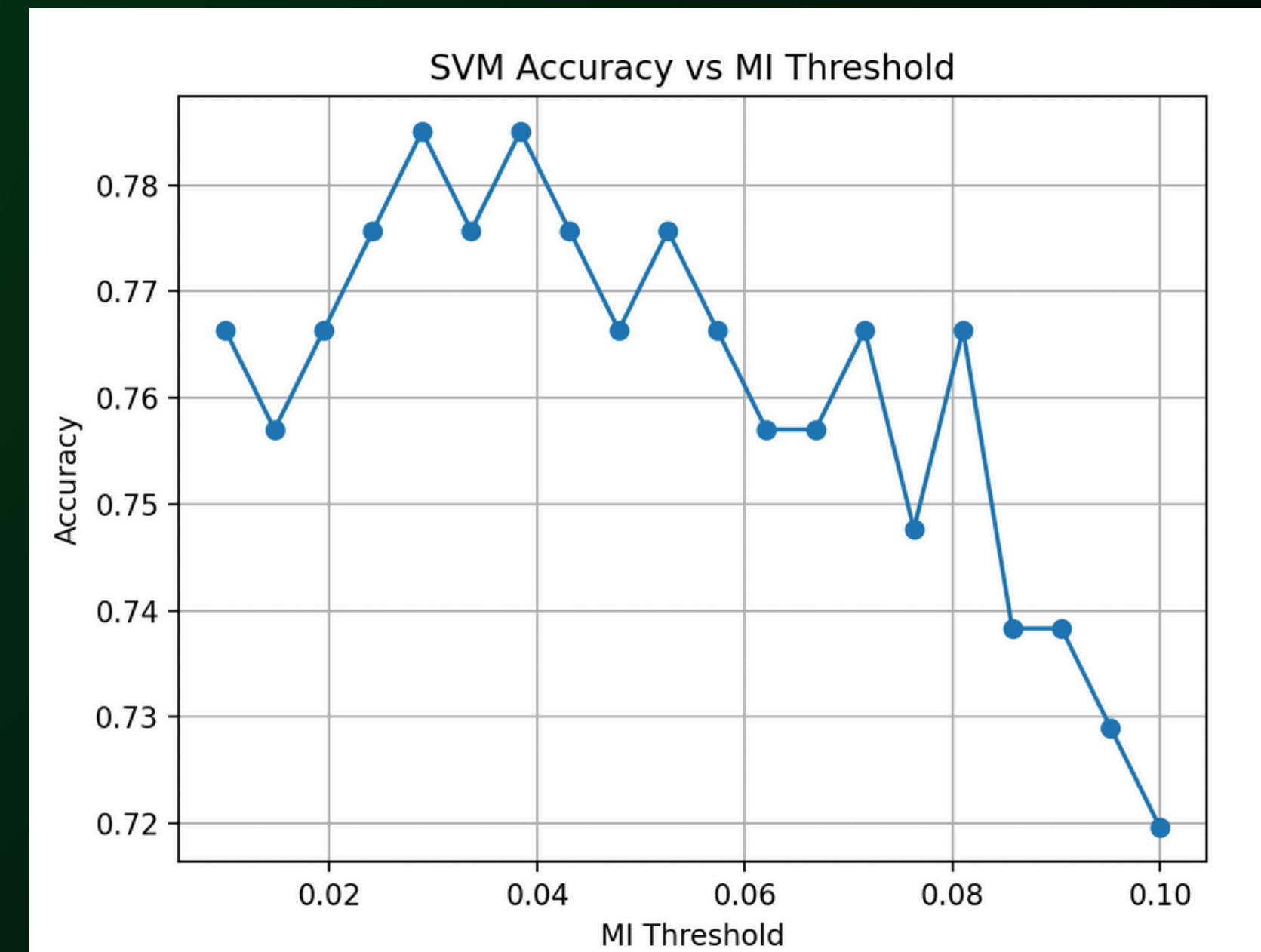


Résultats

On fait varier le MI threshold

Trop haut : peu d'information

Trop bas : trop de bruit



SVM Résultats finaux

Bien meilleurs résultats :

Précision :
71% -> 80%

F1 mean :
69% -> 78%

	precision	recall	f1-score	support
anger	0.89	0.96	0.93	26
anxiety	0.75	0.64	0.69	14
boredom	0.68	0.81	0.74	16
disgust	0.75	0.67	0.71	9
happiness	0.79	0.79	0.79	14
neutral	0.83	0.62	0.71	16
sadness	0.86	1.00	0.92	12
accuracy			0.80	107
macro avg	0.79	0.78	0.78	107
weighted avg	0.80	0.80	0.80	107

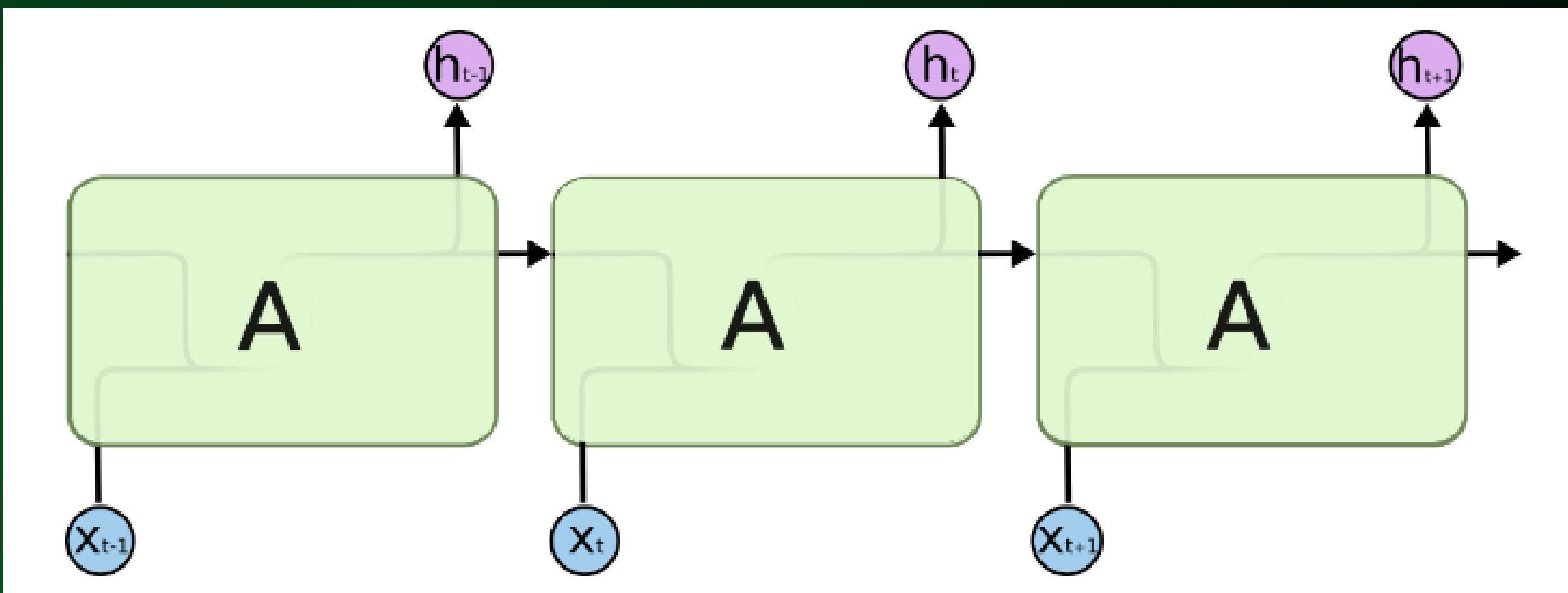
RESEAUX DE NEURONNES AVANCÉS

LSTM

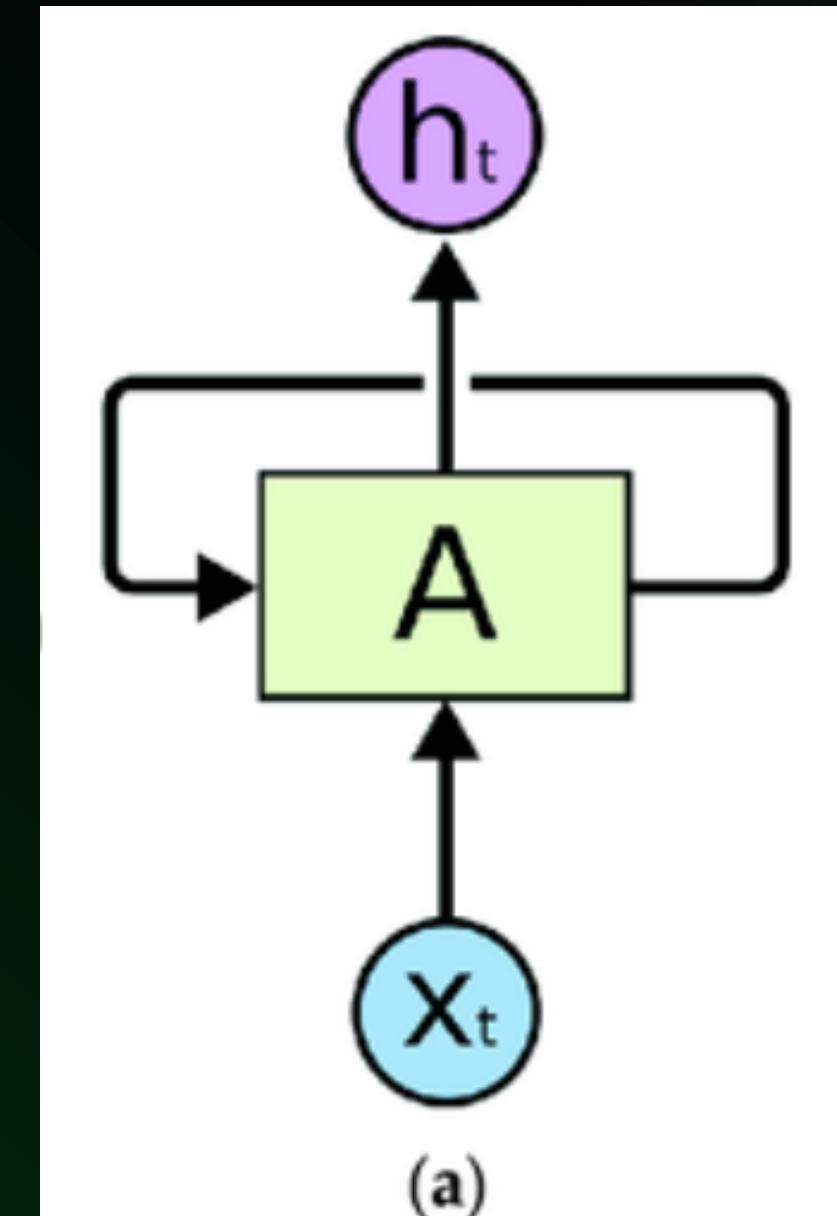
LSTM – Pourquoi s'y intéresser?

Avantages	Inconvénients
<ul style="list-style-type: none">-Réseau de neurones récurrent (RNN)-Comprend les relations temporelles et non linéaires entre les features-Insensible au décalage temporel-Classification supervisée avec plusieurs catégories	<ul style="list-style-type: none">-Mémoire à long terme parfois volatile-Nécessite un grand dataset-Effet boîte noire-Temps d'entraînement

LSTM – Principe de Fonctionnement

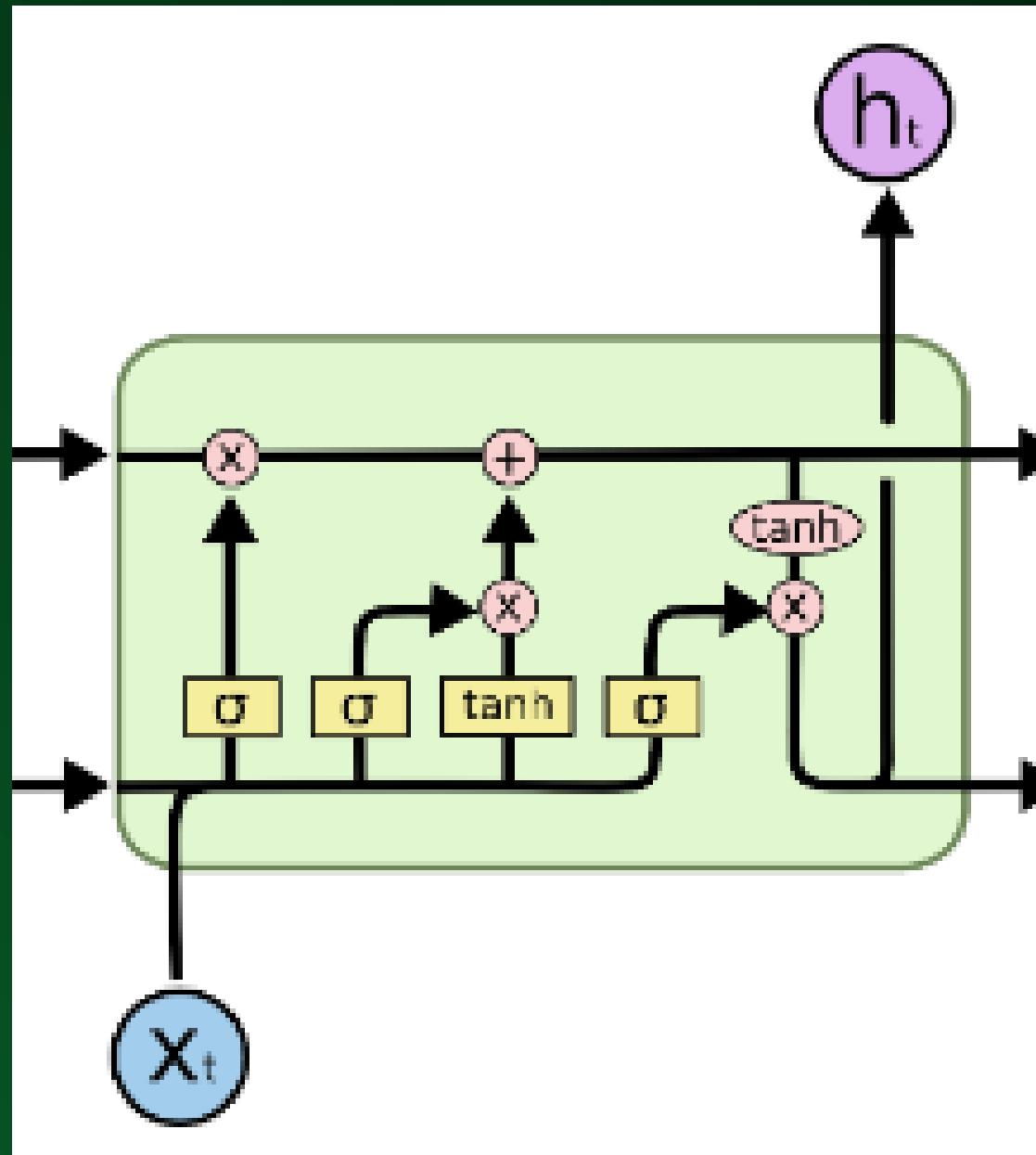


Chaîne d'exécution d'un LSTM à une couche

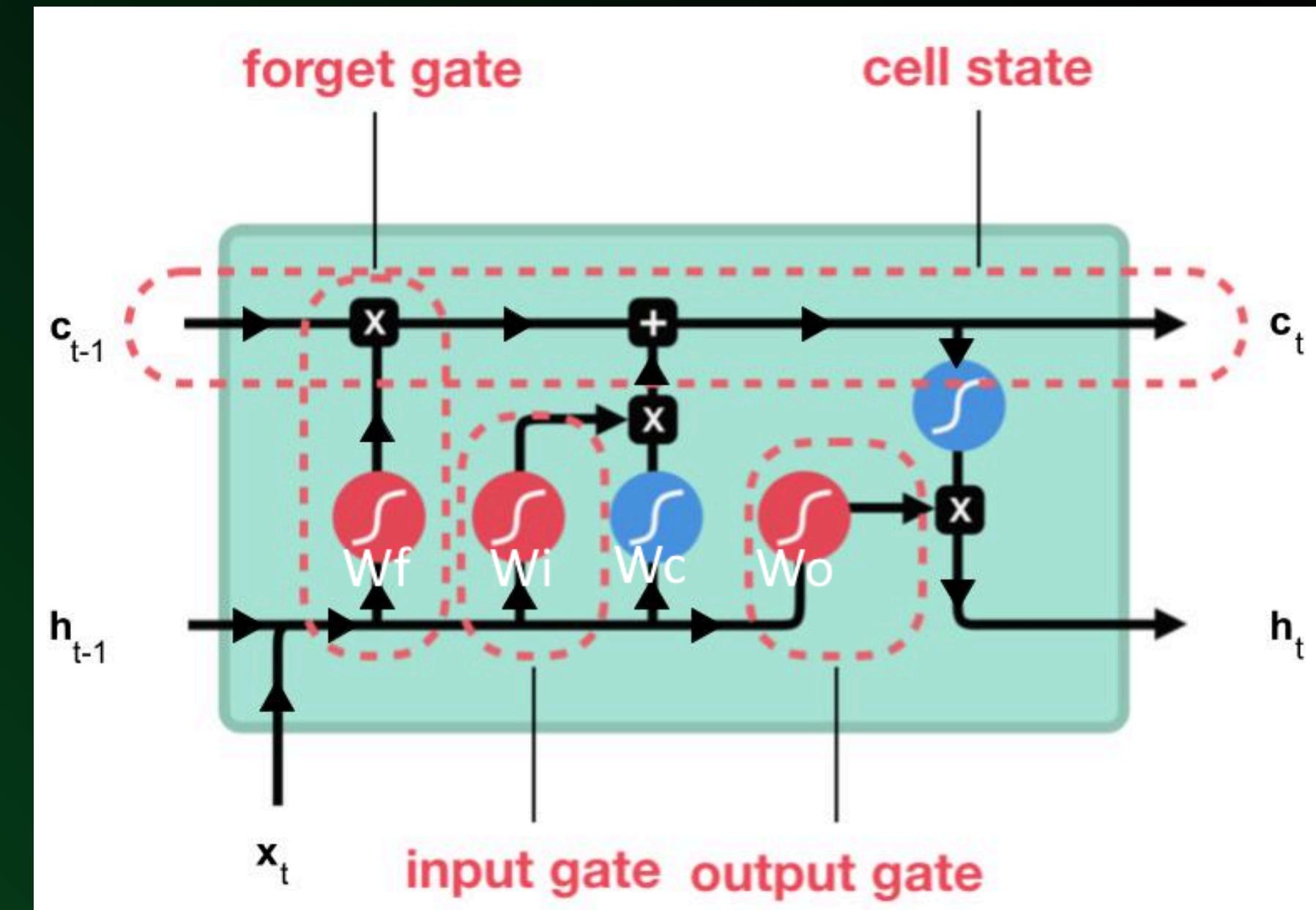


Représentation récurrente

LSTM – Principe de Fonctionnement



L'intérieur d'une cellule



$$P_{\text{total}} = 4 \times (d_c \times (d_{\text{in}} + d_c) + d_c)$$

LSTM - Principe de Fonctionnement

séparation train/test



LSTM – Résultats

	precision	recall	f1-score	support
anger	0.70	0.92	0.79	25
disgust	0.50	0.50	0.50	10
fear	0.38	0.43	0.40	14
happiness	0.40	0.31	0.35	13
neutral	0.57	0.53	0.55	15
sadness	1.00	0.57	0.73	14
accuracy			0.59	91
macro avg	0.59	0.54	0.55	91
weighted avg	0.61	0.59	0.59	91

Confusion Matrix:

[[23 0 1 1 0 0]
[1 5 2 2 0 0]
[3 2 6 3 0 0]
[5 0 4 4 0 0]
[1 3 3 0 8 0]
[0 0 0 0 6 8]]

sur EmoDB

LSTM – Résultats

	precision	recall	f1-score	support
anger	0.66	0.76	0.71	242
disgust	0.38	0.56	0.46	254
fear	0.63	0.30	0.40	264
happiness	0.54	0.46	0.50	236
neutral	0.55	0.48	0.51	212
sadness	0.53	0.62	0.57	281
accuracy			0.53	1489
macro avg	0.55	0.53	0.52	1489
weighted avg	0.55	0.53	0.52	1489

Confusion Matrix:

[[185 29 4 21 1 2]]
[33 143 7 14 24 33]
[16 34 78 44 19 73]
[44 48 20 169 12 3]
[2 56 6 4 101 43]
[2 63 8 9 25 174]]

- Test sur Crema_D entier
- disgust mal détectée

LSTM – Résultats

	precision	recall	f1-score	support
anger	0.74	0.73	0.74	297
disgust	0.54	0.42	0.48	255
fear	0.52	0.50	0.51	278
happiness	0.54	0.58	0.56	259
neutral	0.62	0.64	0.63	229
sadness	0.54	0.63	0.58	248
accuracy			0.59	1566
macro avg	0.58	0.58	0.58	1566
weighted avg	0.59	0.59	0.58	1566

Confusion Matrix:

[[218 18 14 33 12 2]]
[28 168 28 28 26 37]
[13 22 138 50 12 43]
[31 13 33 150 24 8]
[4 13 7 13 146 46]
[0 25 44 5 17 157]]

- Test sur un mélange **NON FILTRE** entre emo_DB et Crema_D
- Difficile de classifier happiness et fear

LSTM – Résultats

	precision	recall	f1-score	support
anger	0.81	0.71	0.76	48
disgust	0.47	0.30	0.37	23
fear	0.39	0.27	0.32	33
happiness	0.36	0.64	0.46	33
neutral	0.72	0.87	0.79	15
sadness	0.84	0.70	0.76	30
accuracy			0.58	182
macro avg	0.60	0.58	0.58	182
weighted avg	0.61	0.58	0.58	182

Confusion Matrix:

[[34 0 6 8 0 0]]
[1 7 2 11 1 1]
[4 4 9 14 1 1]
[3 3 6 21 0 0]
[0 0 0 0 13 2]
[0 1 0 5 3 21]]

- Test sur un mélange FILTRE entre emo_DB et Crema_D
- Difficile de classifier happiness et fear

LSTM - Résultats

	precision	recall	f1-score	support
anger	0.91	0.95	0.93	42
disgust	0.73	0.63	0.68	30
neutral	0.83	0.83	0.83	18
sadness	0.77	0.82	0.79	28
accuracy			0.82	118
macro avg	0.81	0.81	0.81	118
weighted avg	0.82	0.82	0.82	118

Confusion Matrix:

```
[[40  2  0  0]
 [ 4 19  1  6]
 [ 0  2 15  1]
 [ 0  3  2 23]]
```

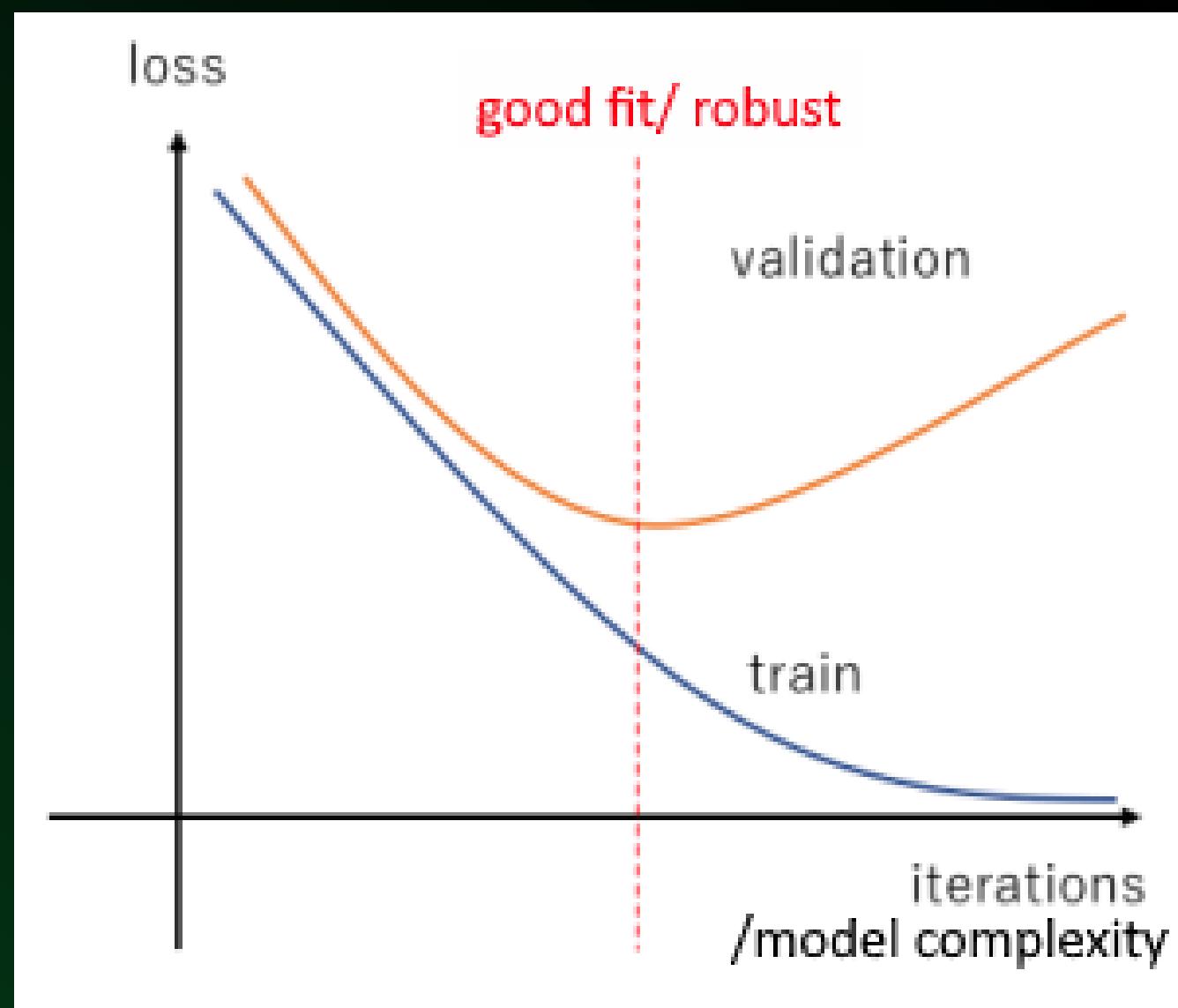
- Test sur un mélange FILTRE entre emo_DB et Crema_D
- Garde seulement 4 emotions
- Meilleure précision

LSTM - Prise de recul sur les résultats

Même avec seulement anger, neutral, sadness et disgust ...

les résultats précédents:

- N_epoch trop faible
- dropout=0.2 trop faible
- taille de batch (32) trop faible
- 4 cellules LSTM
- hidden_layer de 100



Modèle trop complexe qui n'apprend pas assez (manque de robustesse)

LSTM - Modèle fine-tuné

	precision	recall	f1-score	support
anger	0.84	0.93	0.88	28
disgust	0.27	0.50	0.35	6
fear	1.00	0.50	0.67	12
happiness	0.60	0.64	0.62	14
neutral	1.00	0.83	0.91	18
sadness	0.92	0.92	0.92	13
			0.78	91
accuracy			0.78	91
macro avg	0.77	0.72	0.73	91
weighted avg	0.83	0.78	0.79	91

Confusion Matrix:

```
[[26  0  0  2  0  0]
 [ 1  3  0  2  0  0]
 [ 1  2  6  2  0  1]
 [ 3  2  0  9  0  0]
 [ 0  3  0  0  15  0]
 [ 0  1  0  0  0  12]]
```

Disgust problématique

Test sur EmoDB

LSTM - Modèle fine-tuné

	precision	recall	f1-score	support
anger	0.89	0.93	0.91	27
fear	0.67	0.50	0.57	8
happiness	0.71	0.83	0.77	12
neutral	0.83	0.83	0.83	18
sadness	1.00	0.94	0.97	17
accuracy			0.85	82
macro avg	0.82	0.81	0.81	82
weighted avg	0.85	0.85	0.85	82

Confusion Matrix:

[[25 0 2 9 0]
[0 4 2 2 0]
[1 1 10 0 0]
[2 1 0 15 0]
[0 0 0 1 16]]

EmoDB sans disgust

LSTM - Modèle fine-tuné

	precision	recall	f1-score	support
anger	0.73	0.70	0.71	242
disgust	0.57	0.52	0.54	254
fear	0.48	0.58	0.52	264
hapiness	0.51	0.54	0.52	236
neutral	0.63	0.66	0.65	212
sadness	0.59	0.49	0.53	281
accuracy			0.58	1489
macro avg	0.59	0.58	0.58	1489
weighted avg	0.58	0.58	0.58	1489

Confusion Matrix:

```
[[169  16  19  29   8   1]
 [ 21 131  33  28  15  26]
 [  8   13 154  39  10  40]
 [ 28   16  46 128  15   3]
 [  1   16  13  17 140  25]
 [  4   36  59  12  33 137]]
```

CremaD seul

LSTM - Modèle fine-tuné

	precision	recall	f1-score	support
anger	0.75	0.66	0.70	297
disgust	0.62	0.42	0.50	255
fear	0.47	0.54	0.50	278
happiness	0.51	0.60	0.55	259
neutral	0.57	0.63	0.60	229
sadness	0.56	0.59	0.57	248
accuracy			0.57	1566
macro avg	0.58	0.57	0.57	1566
weighted avg	0.58	0.57	0.57	1566

Confusion Matrix:

[[196 18 24 49 7 3]
[23 106 27 24 33 42]
[14 11 149 42 19 43]
[23 19 41 155 16 5]
[3 5 26 28 145 22]
[1 11 50 5 35 146]]

Dataset fusionné et filtré

Pour la même taille de modèle: la différence de langage importe

LSTM - Références

- **LSTM bidirectionnel avec attention** : *Riccardo Cantini* a utilisé un LSTM bidirectionnel avec attention, obtenant 90 % de précision pour la détection des émotions, contre 75 % sans attention. (EmoDB)
- **Reconnaissance des émotions en temps réel** : *MeidanGR* a développé un système LSTM pour la reconnaissance des émotions en temps réel avec 87 % de précision sur des fichiers audio. (Ravdess)
- **Reconnaissance multimodale des émotions** : *Ege Kesim et al.* ont utilisé un modèle LSTM combiné à un Transformer pour la reconnaissance des émotions à partir de données multimodales, obtenant un score de précision de 69% (CREMA-D)

TRANSFORMER – Pourquoi s'y intéresser?

Avantages	Inconvénients
<p>Très bonne précision: capture des relations complexes dans les données.</p> <p>Robustesse: Résilient face au bruit dans les données grâce aux mécanismes d'attention</p>	<p>Nécessite un grand dataset</p> <p>Effet boîte noire</p> <p>Longueur d'exécution : Consommation de ressources computationnelles</p>

TRANSFORMER – Fonctionnement

Avantages	Inconvénients
<p>Très bonne précision: capture des relations complexes dans les données.</p> <p>Robustesse: Résilient face au bruit dans les données grâce aux mécanismes d'attention</p>	<p>Nécessite un grand dataset</p> <p>Effet boîte noire</p> <p>Longueur d'apprentissage: Consommation de ressources computationnelles</p>

CONCLUSION

Comparaison de nos modèles et de la littérature

Modèle	EMO-DB	CREMA D
SVM	83 / 79	60 / 57
Random Forest	84 / 77	65 / 54
LSTM découpée	90 / 85	69 / 57

OUVERTURE

1. Exécution en temps réel
2. Datasets plus larges que seulement la parole
3. Commercialiser le produit

MERCI !



ANNEXE & BIBLIOGRAPHIE

Caractéristiques extraites par OpenSMILE

Catégorie	LLDs
Fréquence	F0 (pitch), Jitter
Amplitude	Shimmer
Rapport	Harmonic-to-Noise Ratio (HNR)
Énergie	Loudness, Energy, Zero-Crossing Rate (ZCR)
Coefficients cepstraux	MFCC 1, MFCC 2, MFCC 3, MFCC 4
Spectral	Spectral Centroid, Spectral Flux, Spectral Entropy, Spectral Variance, Spectral Skewness, Spectral Kurtosis, Spectral Slope, Spectral Flatness, Spectral Sharpness, Spectral Harmonicity
Formants	Formants 1, Formants 2, Formants 3
Largeur de bande des formants	Formant Bandwidth 1, Formant Bandwidth 2, Formant Bandwidth 3

Bibliographie

[1]

Two-layer fuzzy multiple random forest for speech emotion
recognition in human-robot interaction

Luefeng Chen a,b, Wanjuan Su a,b, Yu Feng a,b, Min Wu a,b, *, Jinhua She c,
Kaoru Hirota

[2]

T1 - Improving Speaker-Dependency/Independency of Wavelet-Based Speech Emotion Recognition

VL -

DO - [10.1007/978-3-031-15191-0_27](https://doi.org/10.1007/978-3-031-15191-0_27)

ER -

[3] S. Yan, L. Ye, S. Han, T. Han, Y. Li and E. Alasaarela, "Speech Interactive Emotion Recognition System Based on Random Forest," 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 2020, pp. 1458-1462, doi: [10.1109/IWCMC48107.2020.9148117](https://doi.org/10.1109/IWCMC48107.2020.9148117).

[4]

CREMA-D: Crowd-sourced Emotional Multimodal Actors Dataset

Houwei Cao 1 , David G Cooper 2 , Michael K Keutmann 3 , Ruben C Gur 4 , Ani Nenkova 5 , Ragini Verma 6

82

Bibliographie

[5] Speech emotion recognition using multimodal feature fusion with machine learning approach Sandeep Kumar
Panda¹ · Ajay Kumar Jena² · Mohit Ranjan Panda² · Susmita Panda

[7] An analysis of large speech models-based
representations for speech emotion recognition

Adrian Bogdan ST ^ANEA, Vlad STRILET, CHI, Cosmin STRILET, CHI, Adriana STAN

[8] Speech Emotion Recognition Using a Multi-Time-Scale Approach
to Feature Aggregation and an Ensemble of SVM Classifiers

Antonina STEFANOWSKA, Sławomir K. ZIELIŃSKI *

Bibliographie

[9] Deep Learning Techniques for Speech Emotion Recognition,
from Databases to Models
Babak Joze Abbaschian * , Daniel Sierra-Sosa and Adel Elmaghreby

[10] :-Ververidis, D., & Kotropoulos, C. (2006). Emotional speech recognition: Resources, features, and methods. *Speech communication*, 48(9), 1162-1181.

Bibliographie

- [11] Colah, C. (2015). Understanding LSTMs. Retrieved from <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [12] Analytics Vidhya. (2022). An Overview on Long Short-Term Memory (LSTM). Retrieved from <https://www.analyticsvidhya.com/blog/2022/03/an-overview-on-long-short-term-memory-lstm/>
- [13] MeidanGR. (n.d.). Speech Emotion Recognition - Realtime. Retrieved from https://github.com/MeidanGR/SpeechEmotionRecognition_Realtimet
- [14] Cantini, R. (n.d.). Speech Emotion Detection. Retrieved from https://riccardo-cantini.netlify.app/post/speech_emotion_detection/
- [15] Speech Emotion Recognition: A Survey. (2023). arXiv:2306.13076. Retrieved from <https://arxiv.org/abs/2306.13076>

Bibliographie

- [16] Speech Emotion Recognition: A Survey. (2023). arXiv:2306.13076. Retrieved from <https://arxiv.org/abs/2306.13076>
- [17] Performance Improvement of Speech Emotion Recognition Systems by Combining 1D CNN and LSTM with Data Augmentation. Retrieved from <https://www.mdpi.com/1424-8220/21/13/4399>
- [18] Speech Emotion Detection. Retrieved from https://riccardo-cantini.netlify.app/post/speech_emotion_detection/