

Détection du mildiou de la pomme de terre par imagerie grâce aux méthodes de Machine Learning

Yasmine BOUCHIBTI

Leslie CIETERS

Meryem GRIMAJ

Sommaire

1

Introduction

2

Le jeux de données

3

Modèles de Deep
Learning

4

Modèles de
Machine Learning

5

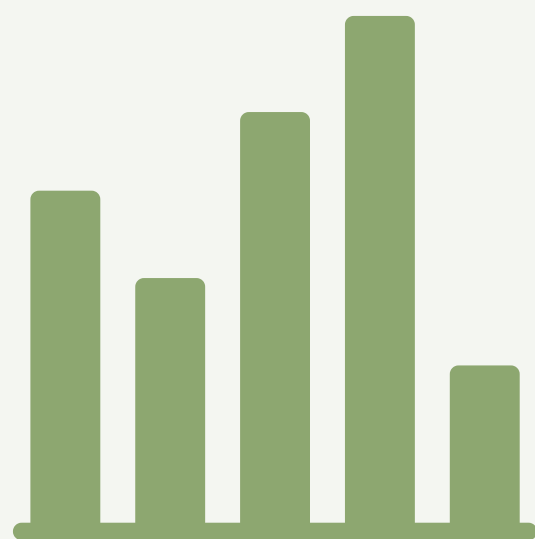
Limites et
perspectives

6

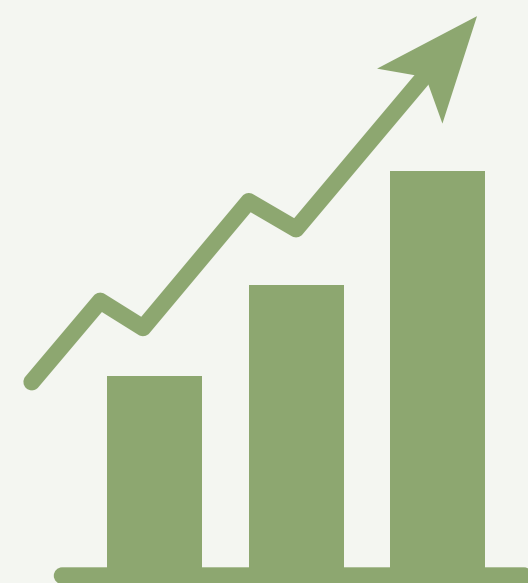
Conclusion

Introduction

- *Alternaria solani* (mildiou précoce)
- *Phytophthora infestans* (mildiou tardif)



Jusqu'à 80% de
pertes

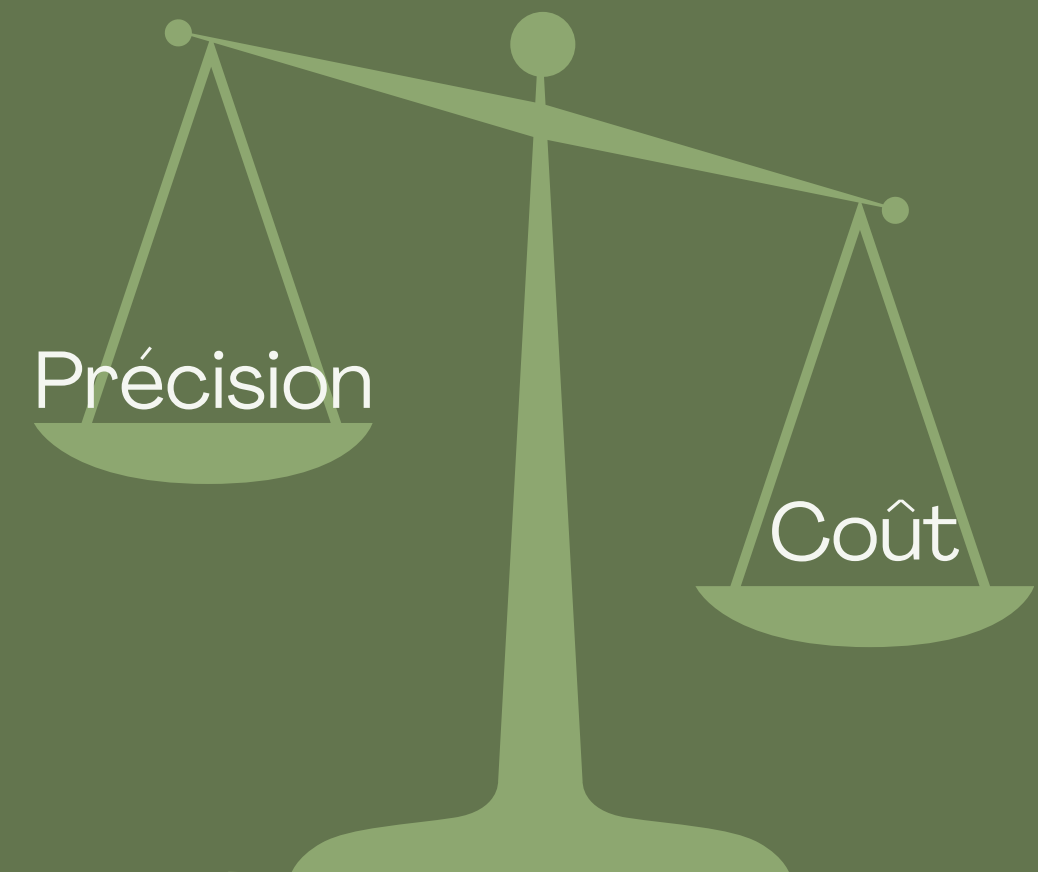
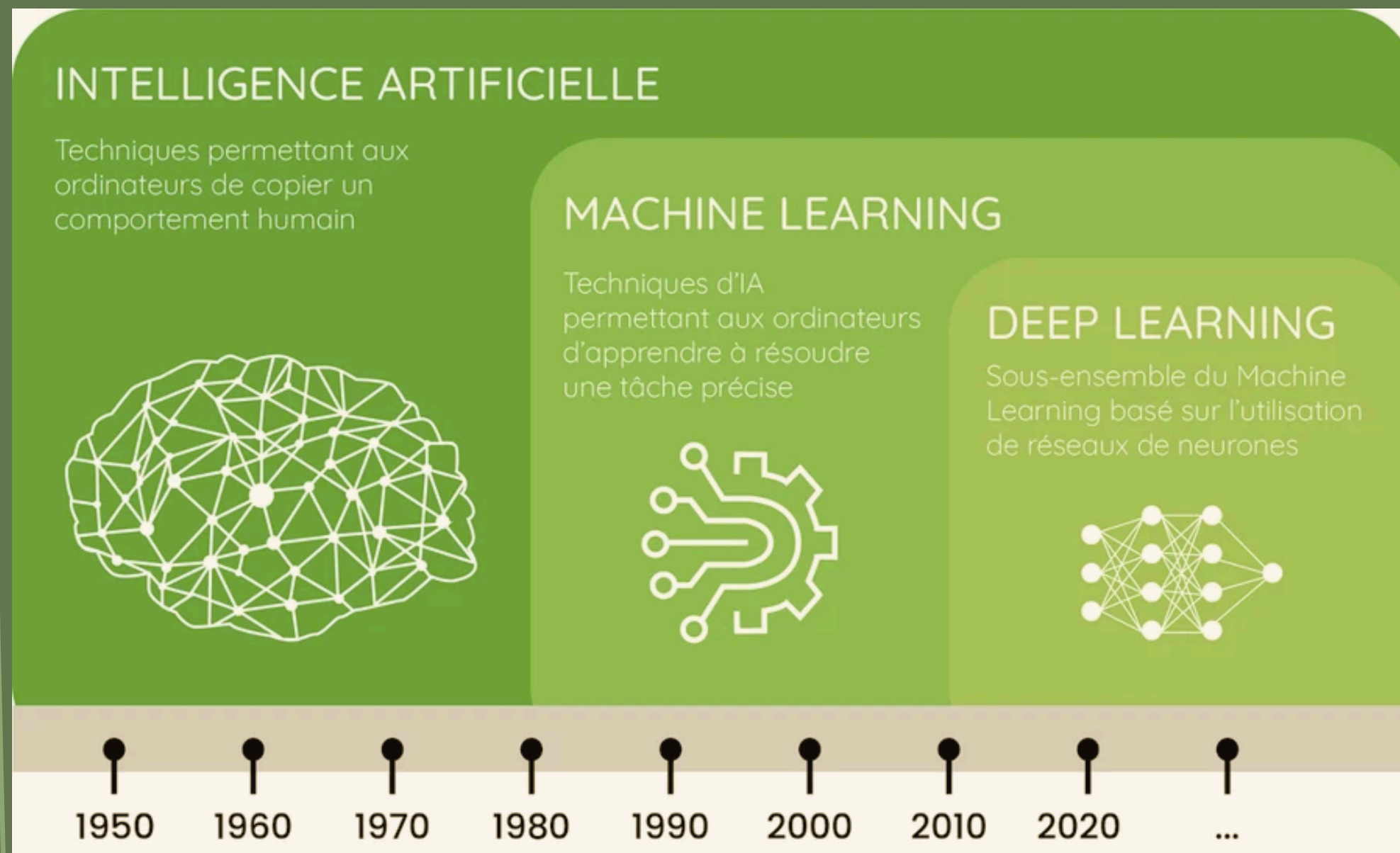


Utilisation de
produits
phytosanitaires



la Grande Famine
1845-1849

Machine Learning et Deep Learning





Le jeu de données

Le jeu de données

Mildiou précoce



Mildiou précoce



Mildiou précoce



Mildiou précoce



Mildiou précoce



Mildiou tardif



Mildiou tardif



Mildiou tardif



Mildiou tardif



Mildiou tardif



Sain



Sain



Sain

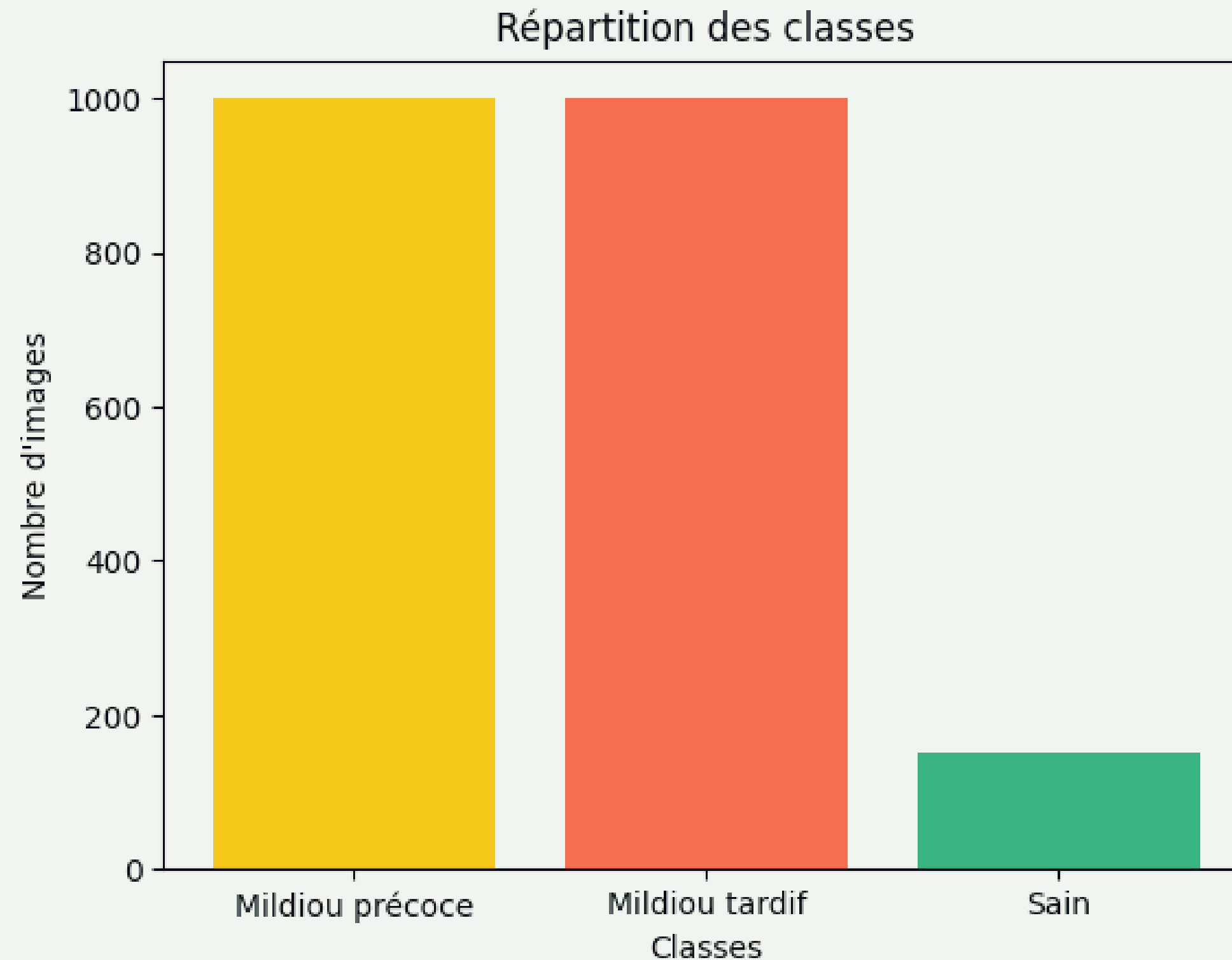


Sain

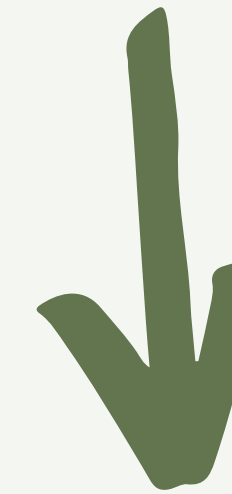


Sain

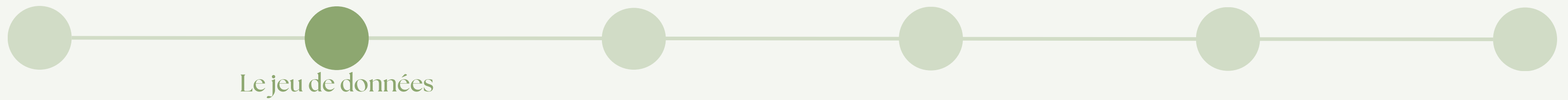




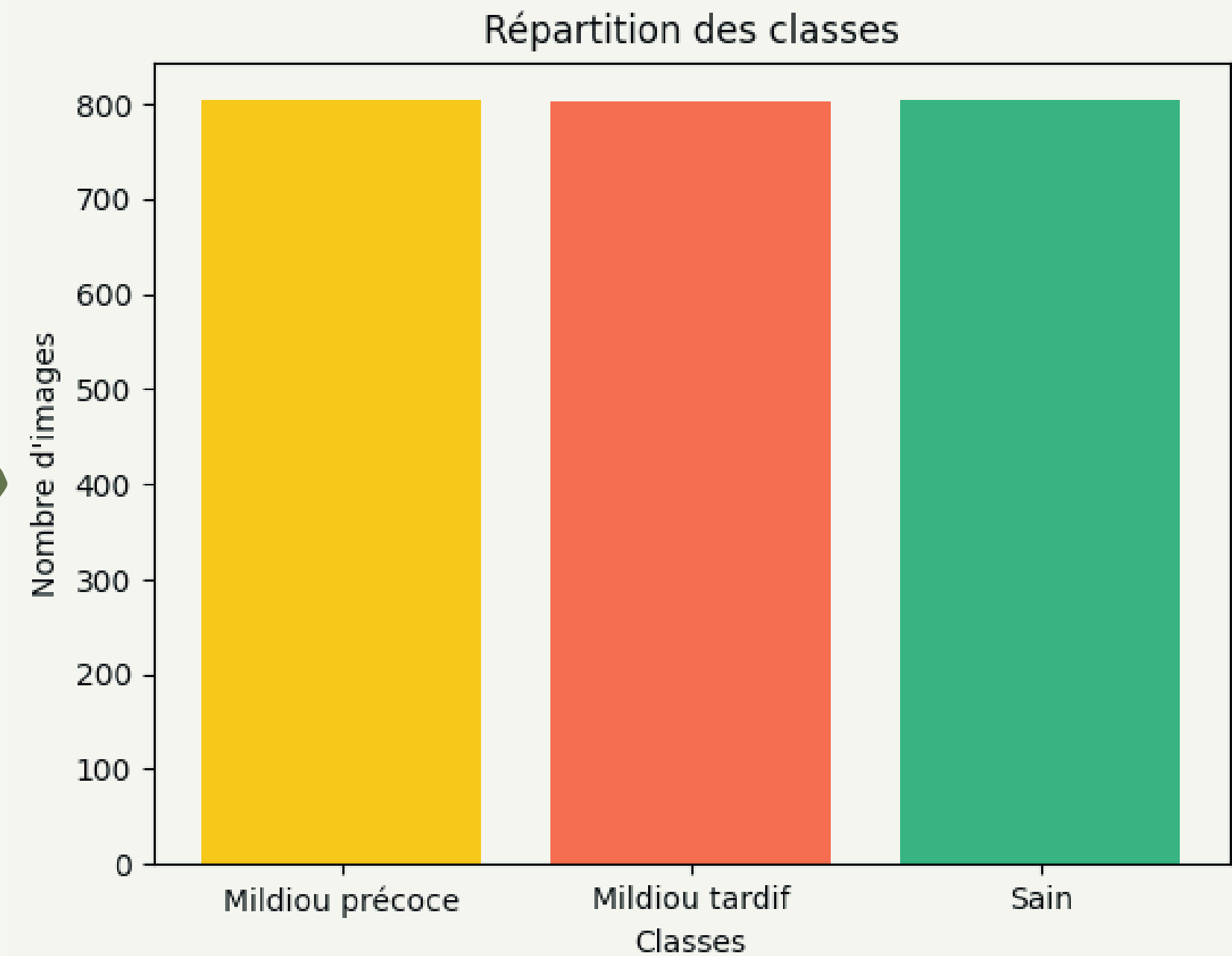
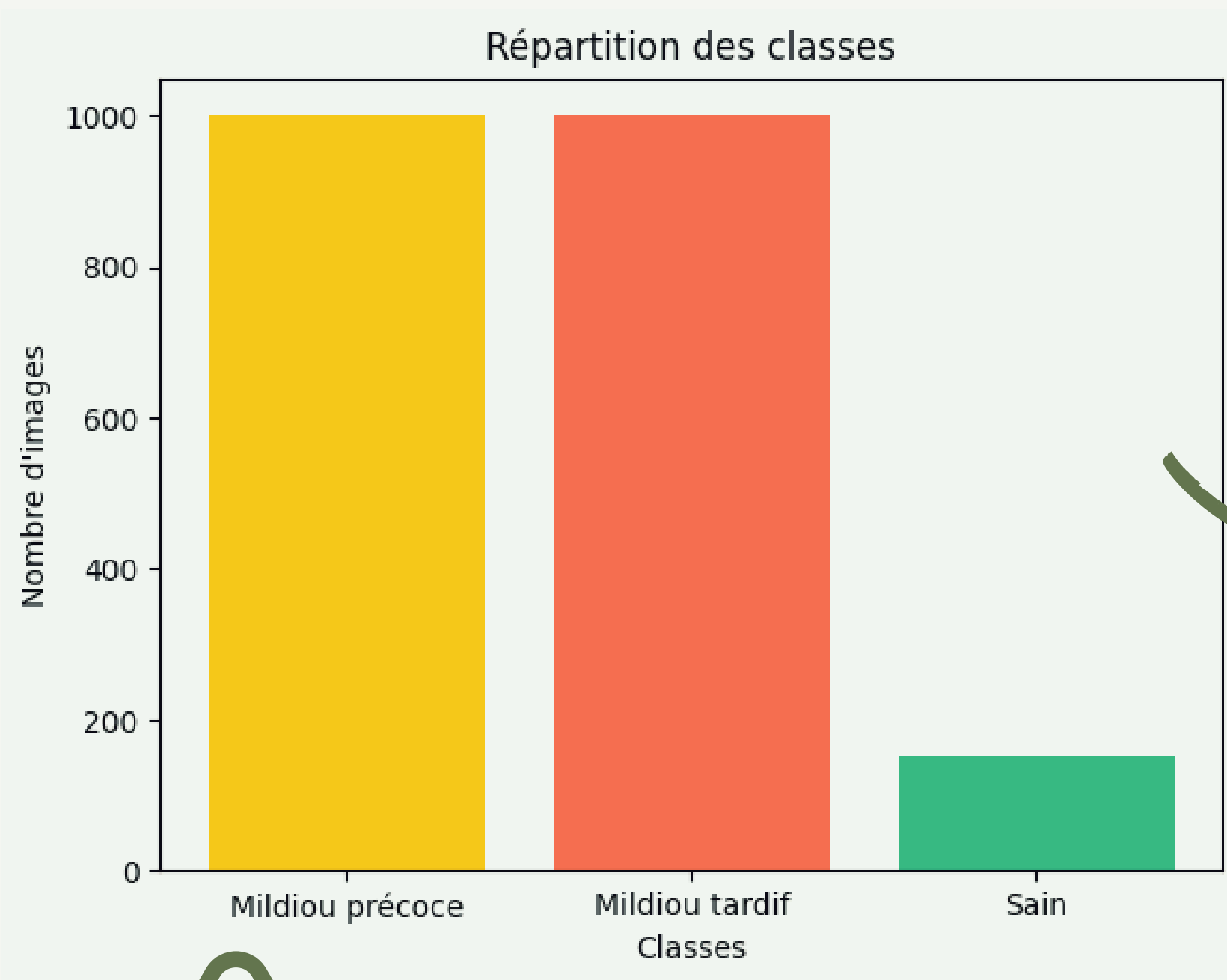
Déséquilibre important des classes



Technique SMOTE pour le
suréchantillonnage de la classe
minoritaire



- Séparation du jeu de données (80/20) pour l'entraînement et la validation



Uniquement sur le jeu de données d'entraînement





Modèles de Machine Learning



Régression logistique

1

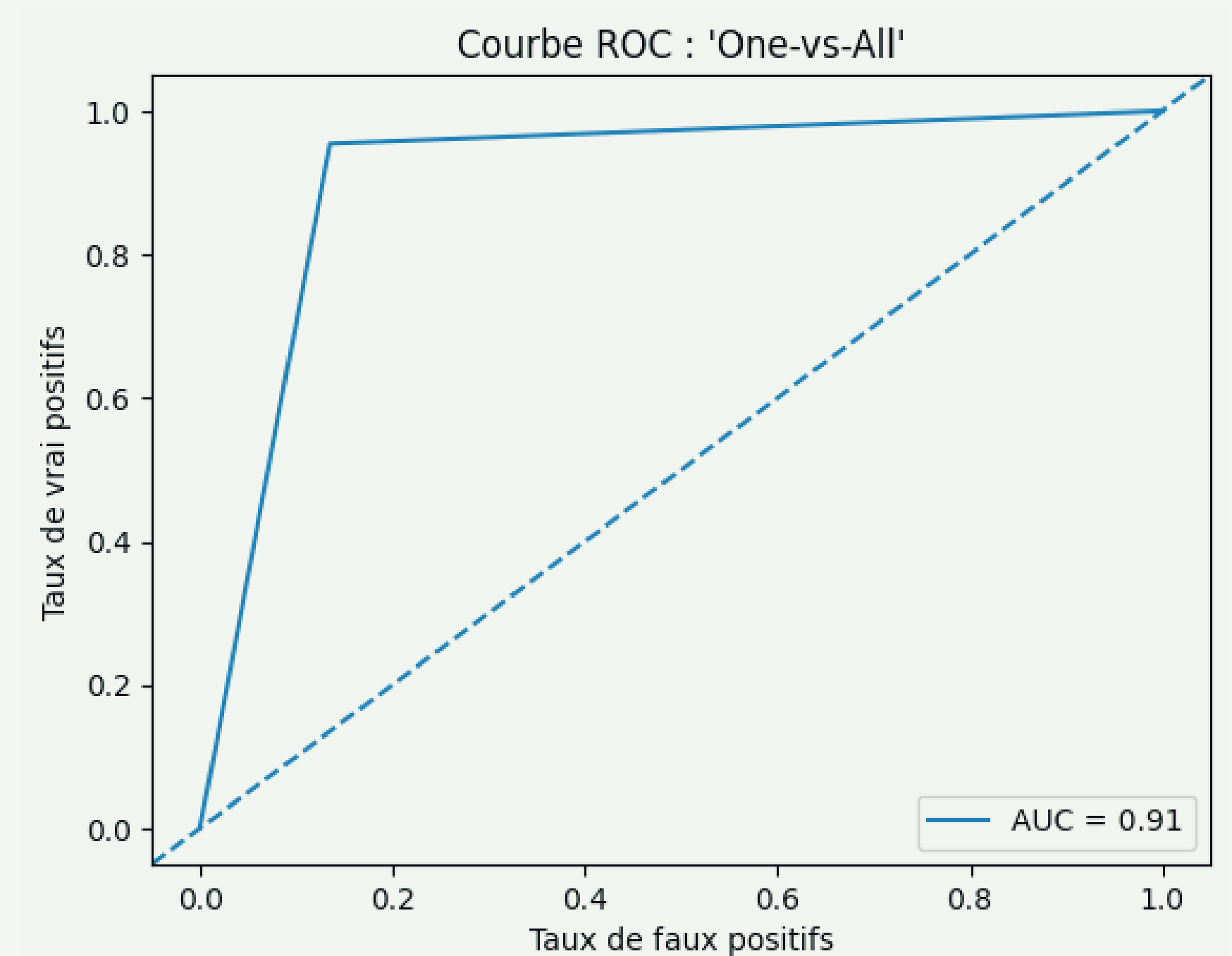
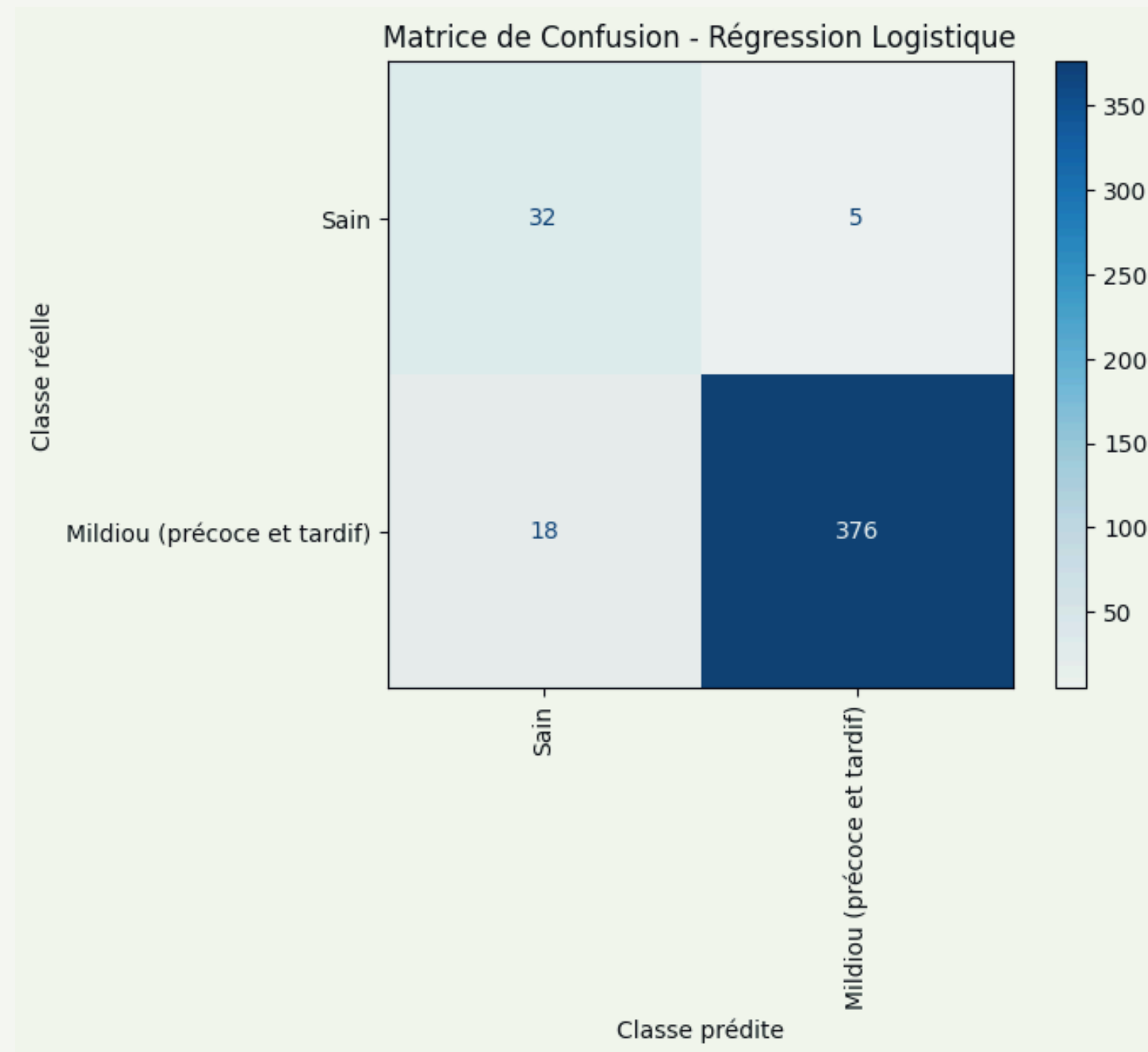
**Méthode
one-vs-all**

2

**Distinction
entre Mildiou
tardif et
précoce**

Résultats

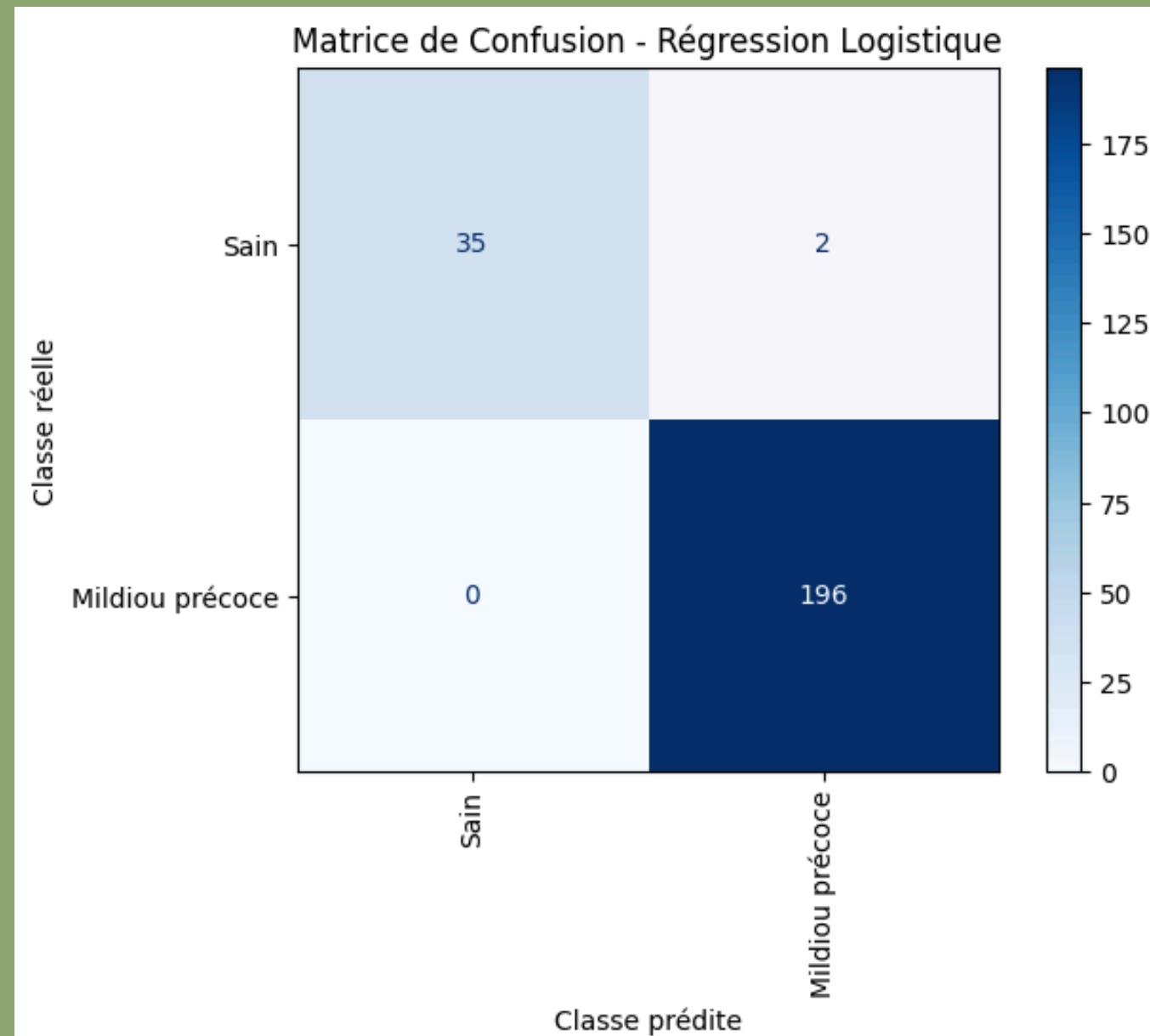
Accuracy = 0.95



Résultats

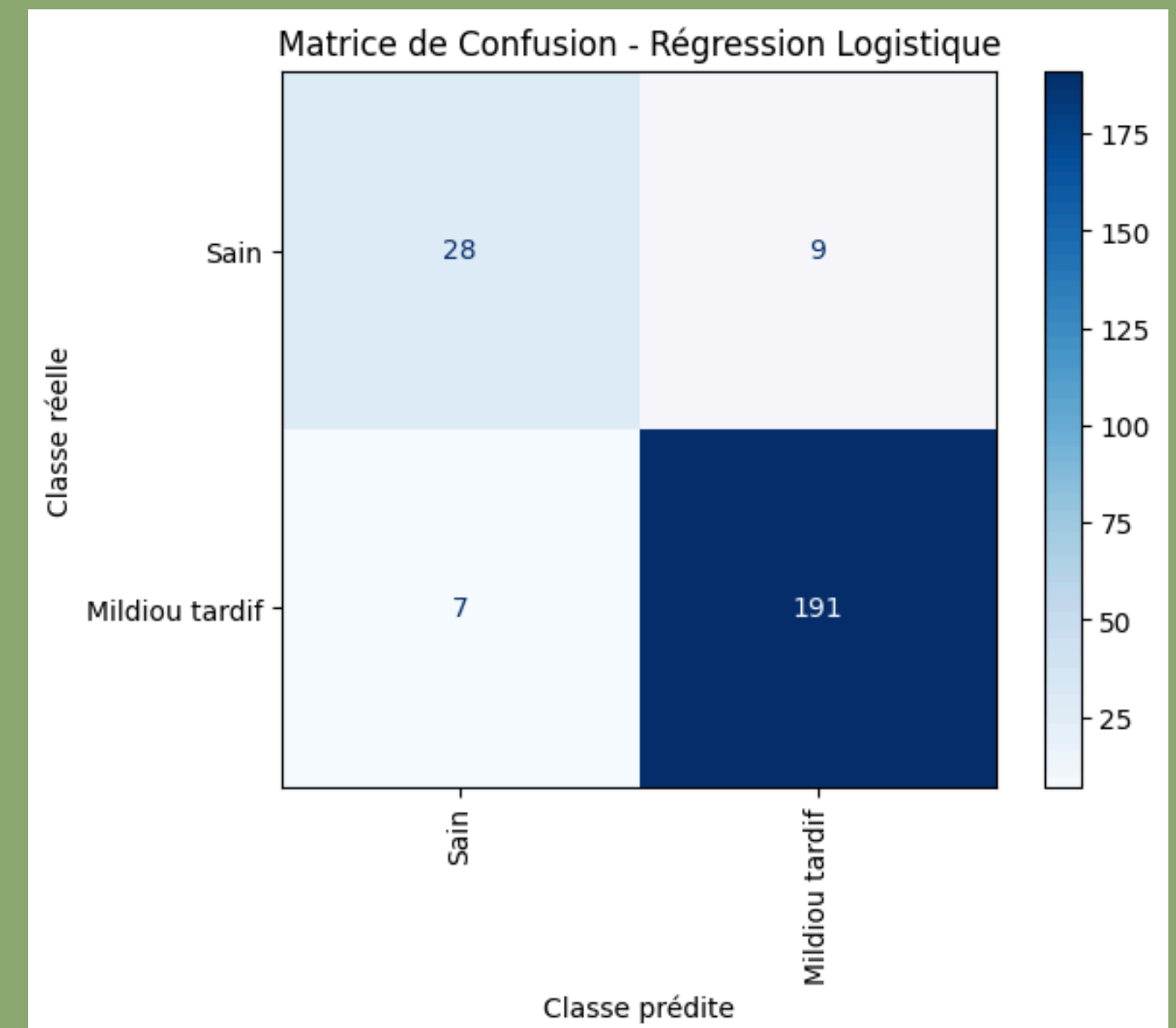
1

Accuracy = 0.99



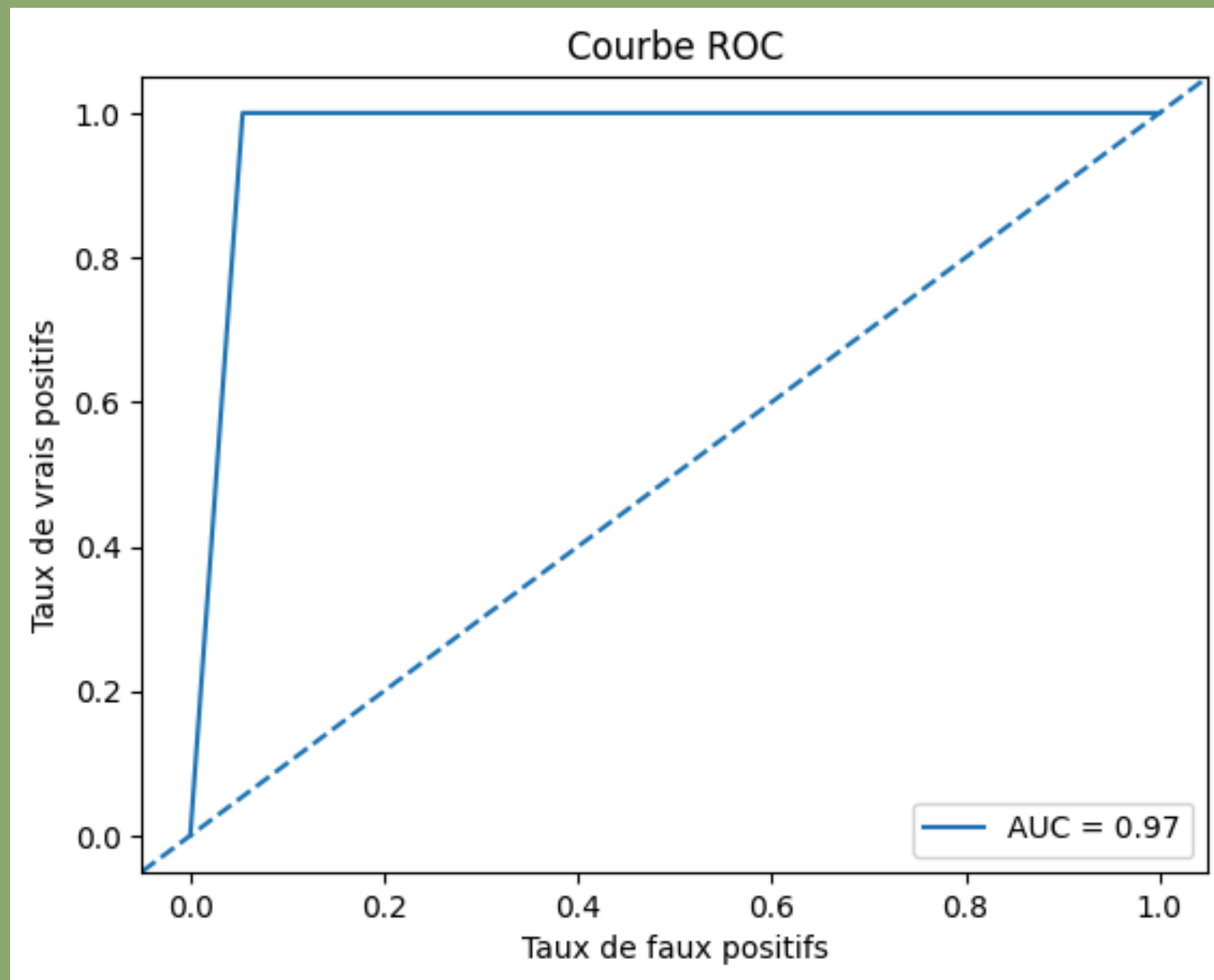
2

Accuracy = 0.93

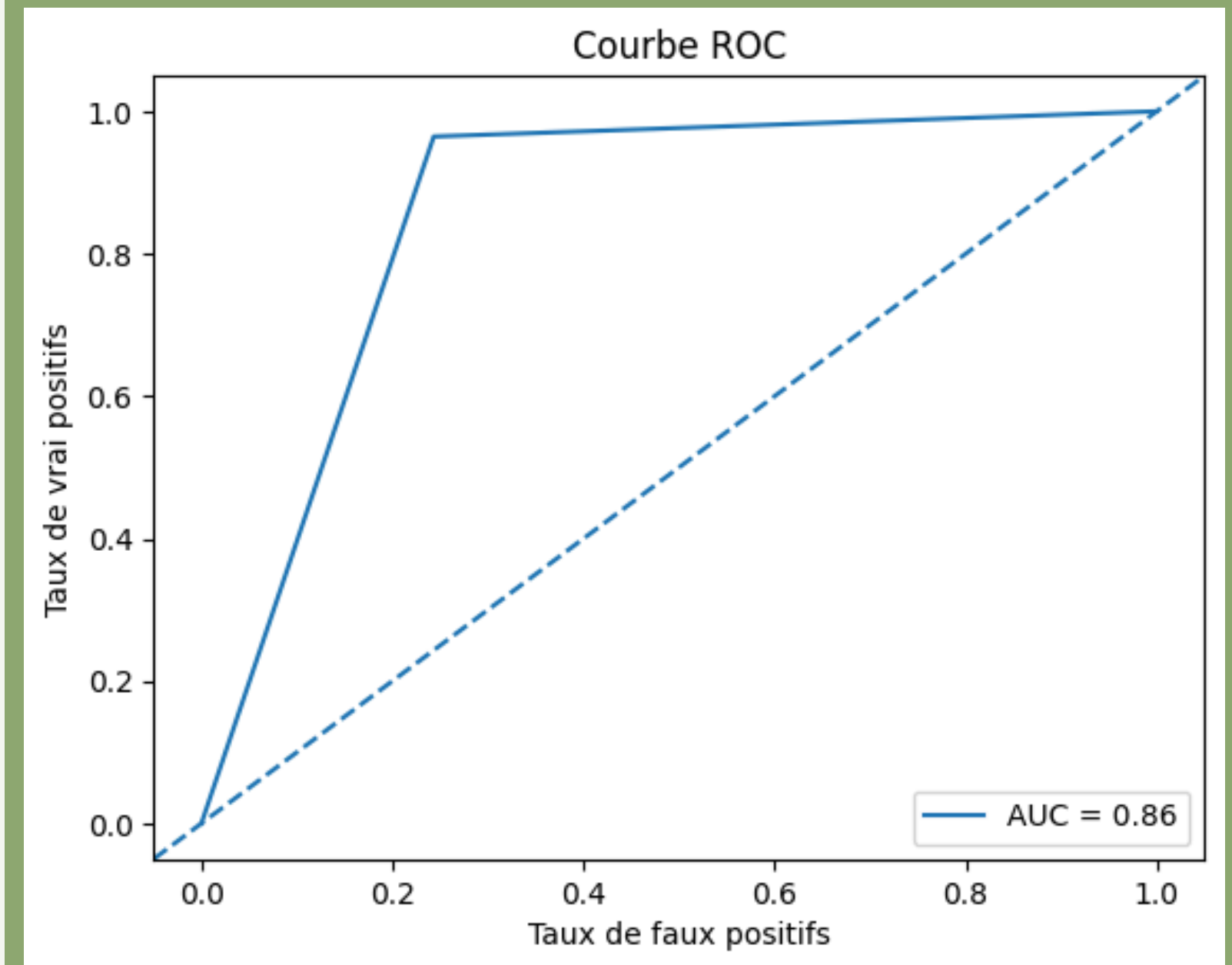


Résultats

1



2



SVM

1

**Choix de la
méthode**

2

**Choix des
hyperparamètres**

Choix de la méthode

Ramener un problème de classification ou de discrimination à un hyperplan

1

2

Efficace dans les espaces de haute dimension

Utilise un sous-ensemble de points d'entraînement dans la fonction de décision (appelé vecteurs de support), il est donc également efficace en mémoire.

3

4

Différentes fonctions du noyau peuvent être spécifiées pour la fonction de décision

Choix des hyperparamètres

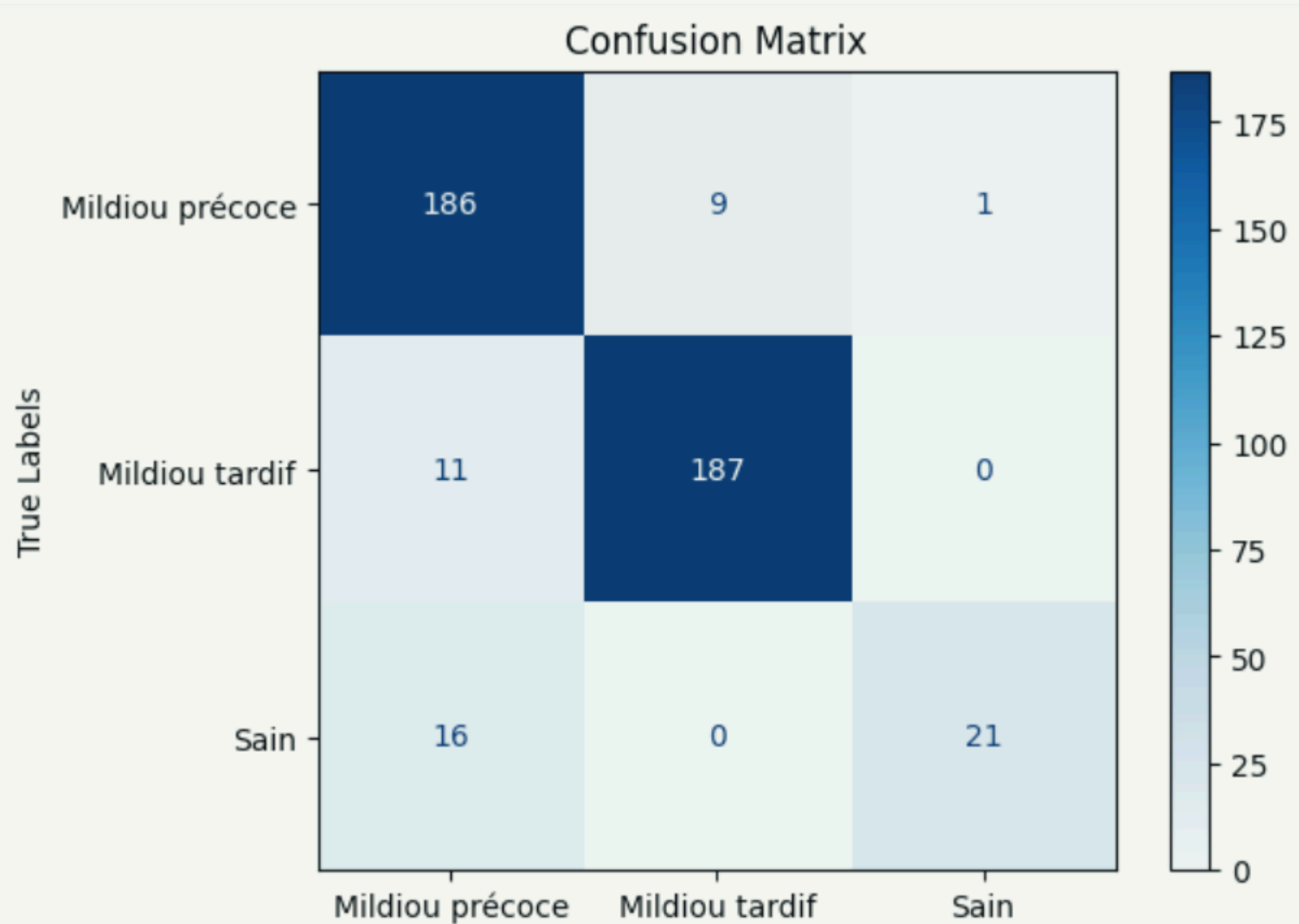
- un contre un pour SVC : des classificateurs sont construits et chacun d'entre eux forme des données à partir de deux classes.
- kernel (noyau) : Linéaire
- decision_function_shape : On choisit d'appliquer la méthode de classificateur un contre le reste au lieu d'un contre un

Résultats

Accuracy: 91.42%

	precision	recall	f1-score	support
0	0.87	0.95	0.91	196
1	0.95	0.94	0.95	198
2	0.95	0.57	0.71	37
accuracy			0.91	431
macro avg	0.93	0.82	0.86	431
weighted avg	0.92	0.91	0.91	431

Modèles de Machine Learning





CONCLUSION POUR LES SVM

Le modèle fonctionne très bien pour les classes majoritaires (0 et 1) avec des précisions, rappels et F1-scores élevés

Random Forest

1

**Choix de la
méthode**

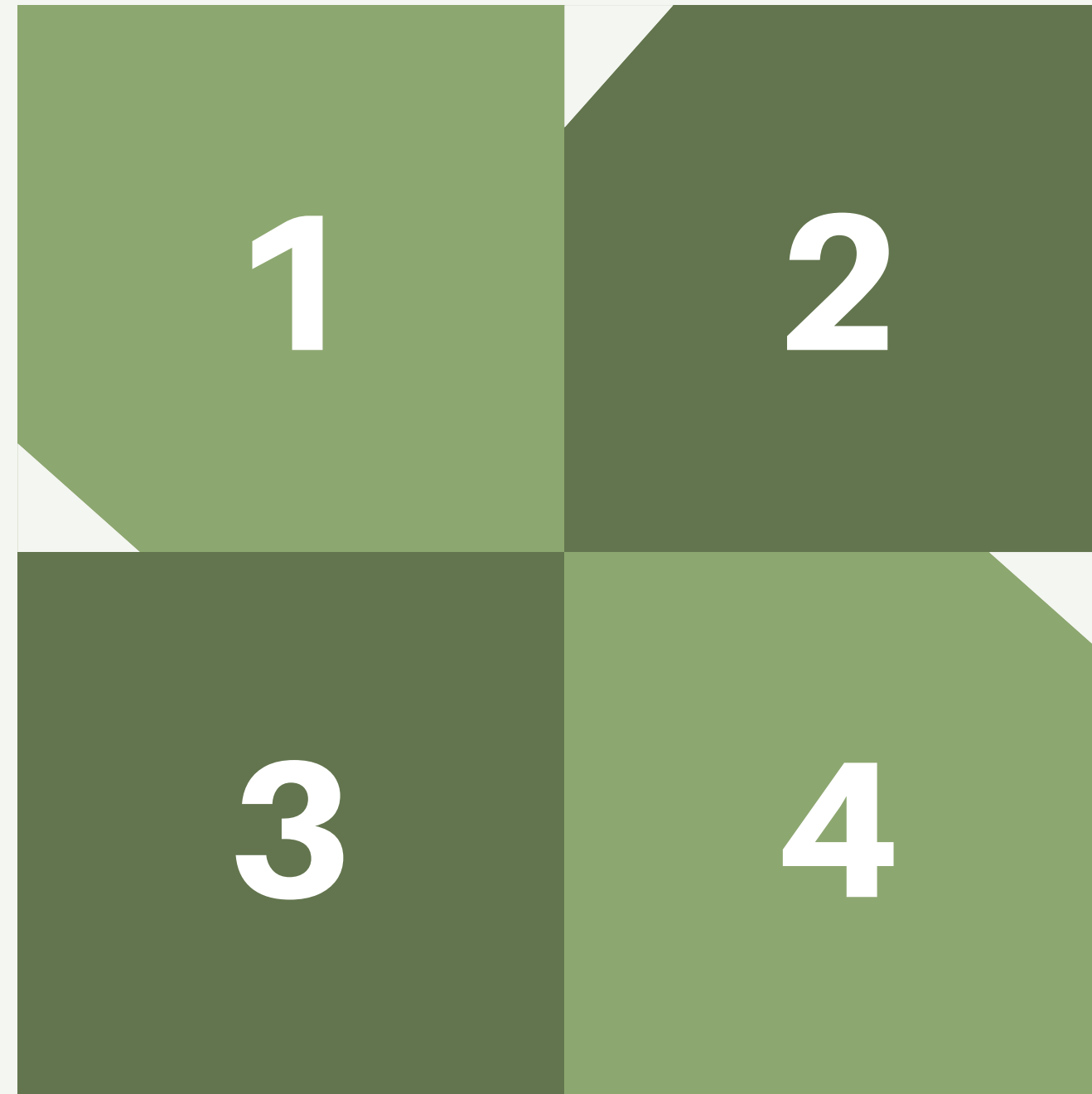
2

**Choix des
hyperparamètres**

Choix de la méthode

Random Forest est également très résistant au bruit dans les données

Solides sur les problèmes de classification grâce à la combinaison de plusieurs arbres de décision.



Contrairement à un modèle de décision unique (comme le `DecisionTreeClassifier`), le Random Forest utilise plusieurs arbres, ce qui réduit la variance du modèle.

Mieux équipé pour traiter des classes déséquilibrées que des modèles plus simples, grâce à sa capacité à effectuer un échantillonnage aléatoire lors de la construction de chaque arbre.

Choix des hyperparamètres

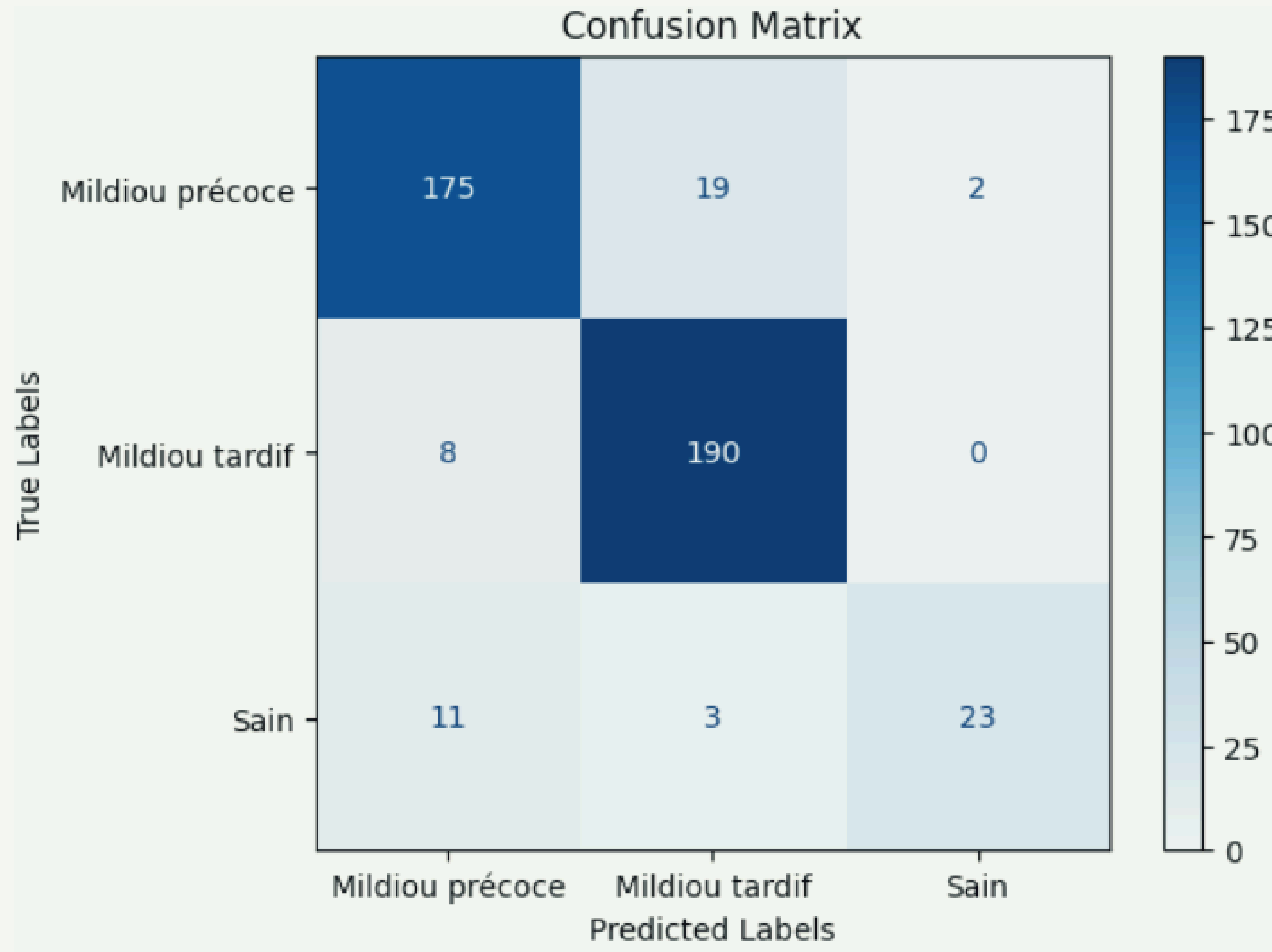
- `n_estimators=100` : 100 arbres de décision se qui améliore la stabilité des prédictions
- `max_features='sqrt'` : Pour chaque division dans un arbre, seulement la racine carrée du nombre total de caractéristiques sera considérée. Cela permet d'accélérer le calcul .

Résultats

Random Forest Accuracy: 0.90

	precision	recall	f1-score	support
0	0.90	0.89	0.90	196
1	0.90	0.96	0.93	198
2	0.92	0.62	0.74	37
accuracy			0.90	431
macro avg	0.91	0.82	0.86	431
weighted avg	0.90	0.90	0.90	431

Modèles de Machine Learning



CONCLUSION POUR LES DECISION TREE

Avec une accuracy de 90% et des précisions et F1-scores élevés pour les classes 0 et 1, le modèle Random Forest se comporte très bien pour les deux premières classes, qui ont plus de données d'entraînement. Le modèle a plus de difficultés à bien identifier la classe 2 (62% de rappel)



Le jeu de données

Imagerie

Le jeu de données

2152 images réparties en
trois classes :

- Mildiou précoce
- Mildiou tardif
- Sain

Modèle CNN

1

**Choix de
l'architecture**

2

**Analyse des
performances
du modèle**

Choix de l'architecture

Modèle CNN

1 couche dense + fonction d'activation Softmax

1 couche dense + fonction d'activation ReLu

1 couche convolutive
Conv2D

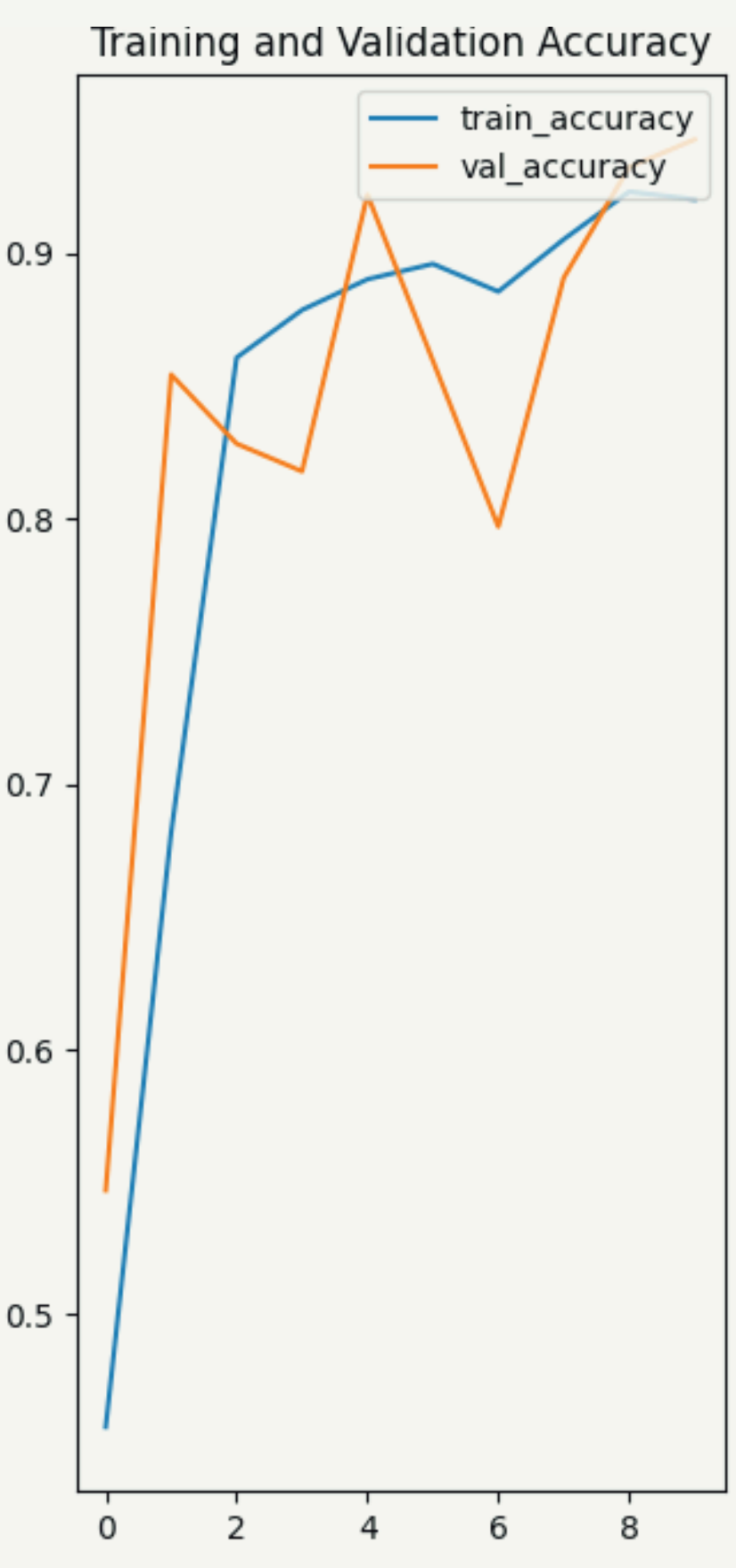
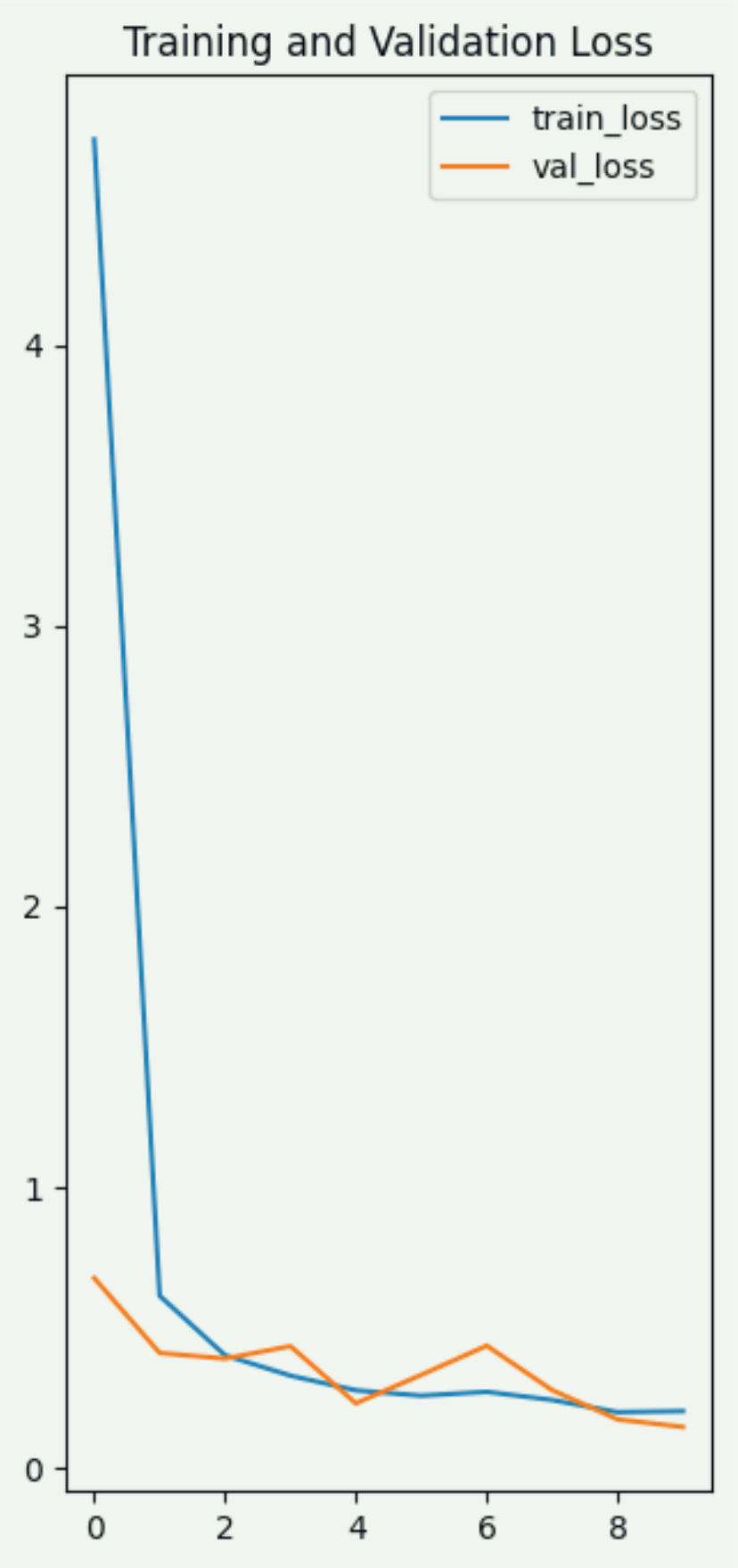
1 couche Flatten

1 couche pooling Pooling2D

CNN

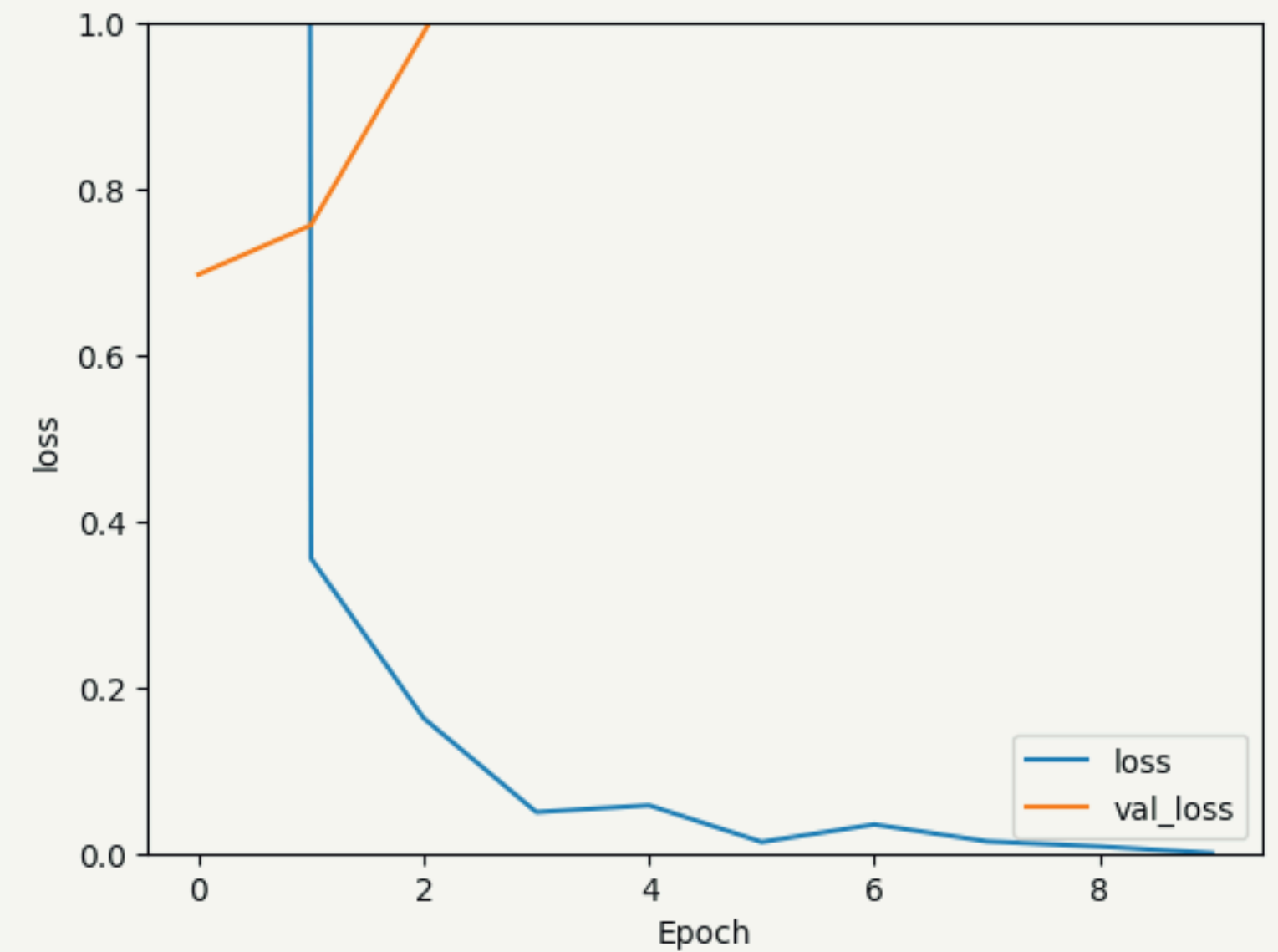
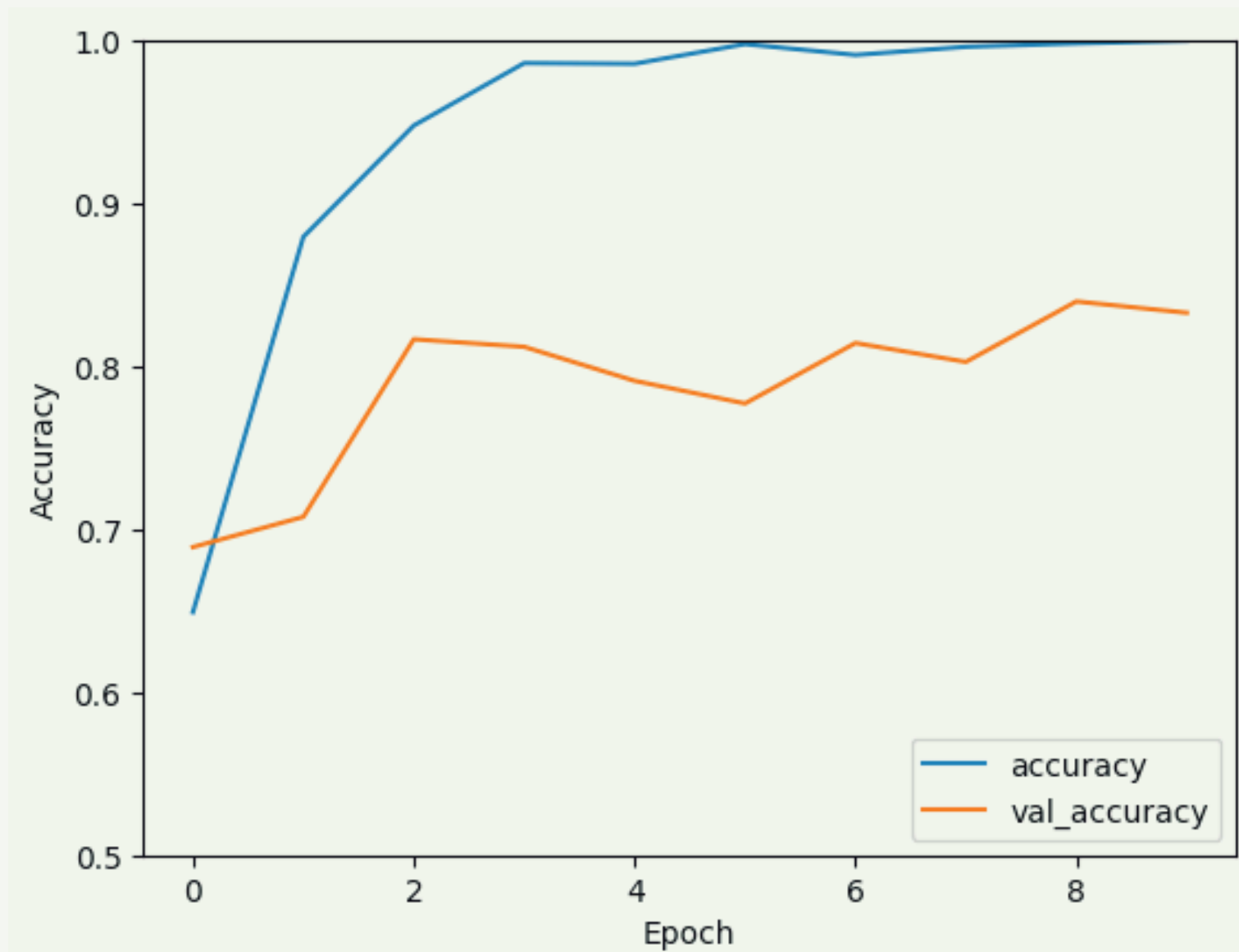
```
graph TD; CNN((CNN)) --> A[1 couche dense + fonction d'activation Softmax]; CNN --> B[1 couche convolutive Conv2D]; CNN --> C[1 couche pooling Pooling2D]; CNN --> D[1 couche Flatten]; CNN --> E[1 couche dense + fonction d'activation ReLu];
```

Classes déséquilibrées

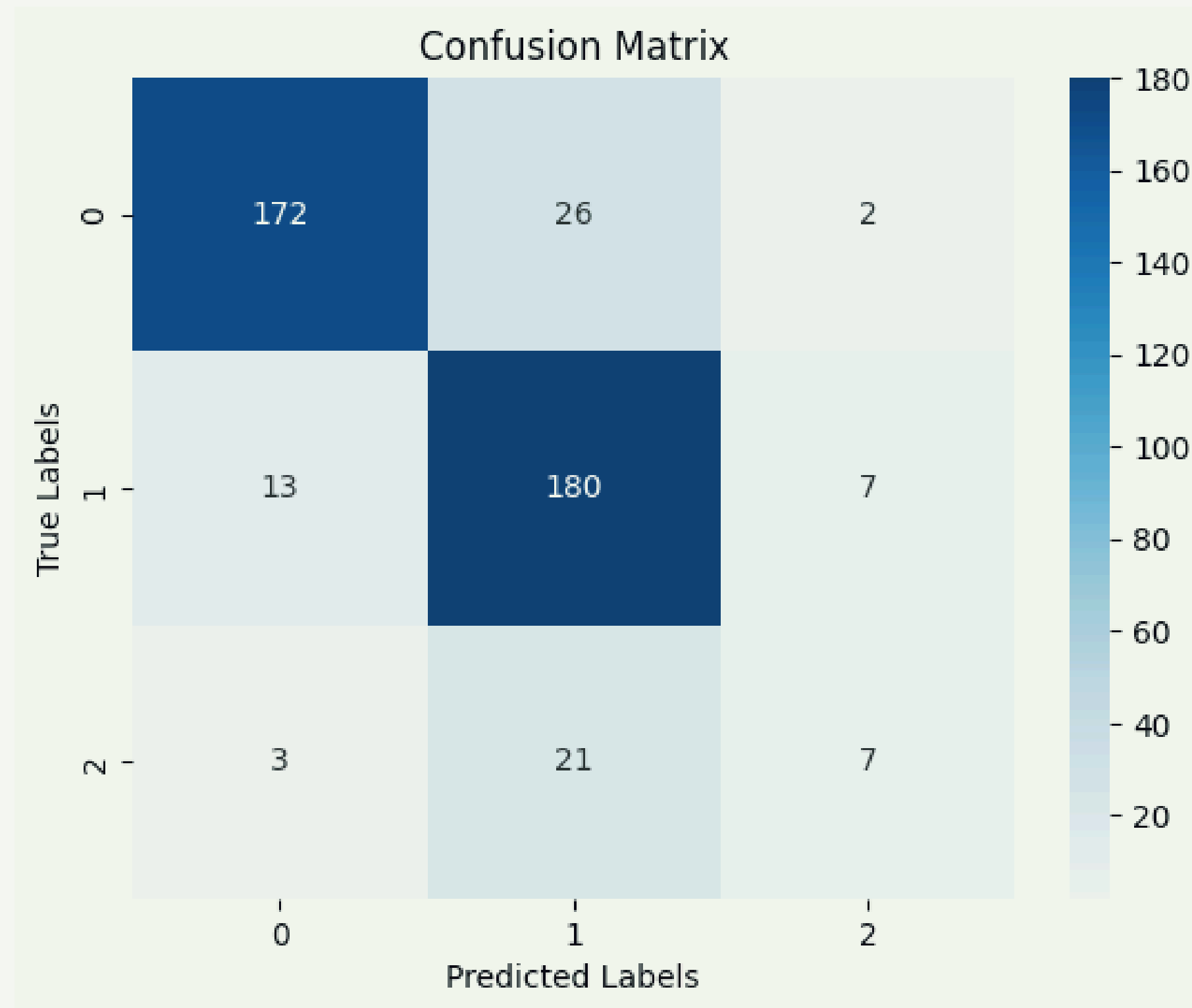


Accuracy : 0.95
Loss: 0.1332

Classes rééquilibrées



Classes rééquilibrées



Accuracy : 0.83

Comparaison avec modèles existants

- Utilisation de la librairie EfficientNet
- Pondération des classes
- Utilisation de 50 epochs

27/27 ————— 1s 49ms/step - accuracy: 0.9584 -
loss: 0.1721
Validation loss: 0.18122921884059906
Validation accuracy: 0.9558139443397522



Discussion

Temps de calcul important, nécessité d'utiliser Google Colab

Capacité de calcul des machines limitées,
modèle de Deep Learning simplifié

Limites

Construction d'un
modèle
hiérarchique entre
les classes

Perspectives



Conclusion

INSTITUT AGRO RENNES-ANGERS

Merci !

Yasmine BOUCHIBTI

Leslie CIETERS

Meryem GRIMAJ