Projet de reconnaissance sonore

**Rapport n°1 : prise en main des caractéristiques et premières mises en place de méthodes**

Auteurs

[**Introduction 3**](#_gs8zflgxjnq4)

[**1. Classification de signaux acoustiques artificiels 4**](#_u4xvab4zk9w5)

[a. Création de la base de données 4](#_mbeu1z3kwo9f)

[b. Description des caractéristiques acoustiques 4](#_kxhehkcyquz4)

[c. Mise en place du/des modèle(s) de ML 4](#_5etwpovrs3ie)

[d. Résultats de classification 4](#_bzs9z9bkwhnz)

[e. Analyse et perspectives 4](#_57q83wq2u5qj)

[**2. Classification de signaux acoustiques réels 5**](#_i01c65k0gbsv)

[a. Cadre de la classification 5](#_dgidgiyjie4q)

[b. Résultats bruts avec le modèle précédent 5](#_g5w1adco9epp)

[c. Analyses et pistes d’amélioration 5](#_fk2j0dm3lnjq)

[d. Résultats améliorés 5](#_yctgt6g94b4m)

[e. Analyse et perspectives supplémentaires 5](#_s0jtdfbcpmdo)

[**3. Mise en place des méthodes d’ensemble 6**](#_3anw889rldvy)

[a. Résultats avec la méthode ??? 6](#_j80xw64yfy81)

[b. Analyse des résultats 6](#_chxxfh8gsmwv)

[c. Améliorations mises en place 6](#_eem971ruxjtg)

[**Conclusion 7**](#_t72kn3unsaod)

# Introduction

L’objectif de ce rapport est de décrire les différentes méthodes mises en place pour travailler le sujet de la reconnaissance sonore au sein de l’entreprise.

Dans une première partie, nous expliquerons les choix concernant les données artificiellement créées et les méthodes mises en place afin de permettre la classification des signaux acoustiques dans deux classes : sinus vs bruits blancs.

Dans une seconde partie, nous verrons comment généraliser les modèles et les caractéristiques acoustiques utilisés dans la première partie pour travailler un problème plus complexe : la classification de signaux sonores d’alarmes de véhicules à partir d’une base de données de signaux acoustiques réels.

Nous analyserons en détail les problèmes pouvant intervenir avant de conclure sur nos recommandations quant à l’utilisation des différentes méthodes et caractéristiques pour la reconnaissance sonore.

# Classification de signaux acoustiques artificiels

## Création de la base de données

Pour la création de la base de données, nous avons fait le choix de faire varier à la fois la fréquence des sons, mais aussi leur amplitude. Nous aurions pu nous contenter de la fréquence, mais nous recherchions à mettre en place un modèle qui saurait distinguer des sons d’amplitudes différentes.

Pour la variations de l’amplitude, nous avons choisi de faire varier l’amplitude par de 0,1 à 1, par pas de 0,1.

A chaque palier nous créons 10 sons différents avec des fréquences aléatoires. Nous obtenons donc au final 100 sinus, et nous créons par la même occasion 100 bruits blancs générés aléatoirement.

## Description des caractéristiques acoustiques

La différence entre un sinus et un bruit blanc se situe dans leur régularité. Un sinus est un signal régulier, tandis qu’un bruit blanc est un signal à la fréquence aléatoire.

## Mise en place du/des modèle(s) de ML

Pour ce qui est du choix du modèle de Machine Learning, nous avons opté pour le Support Vector Machine (SVM).

Comme la différence entre les différents sons est assez évidente, nous avons décidé d’opter pour des hyperparamètres simples :

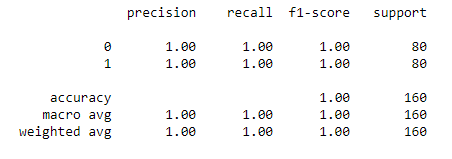
C, cet hyperparamètre toujours positif est le paramètre de régularisation. Pour faire simple, il décide de la marge que l’on va utiliser entre notre hyperplan et les points. Un C très petit peut mener à des mauvaises classification même dans le cas où nos données sont linéairement séparables. Ici nous avons décidé d’utiliser C = 10.

Pour le kernel, nous avons opté pour le kernel linéaire, car les données sont bien découpées et qu’il n’est donc pas nécessaire d’utiliser un kernel plus complexe.

Les autres hyperparamètres ont été laissés par défaut.

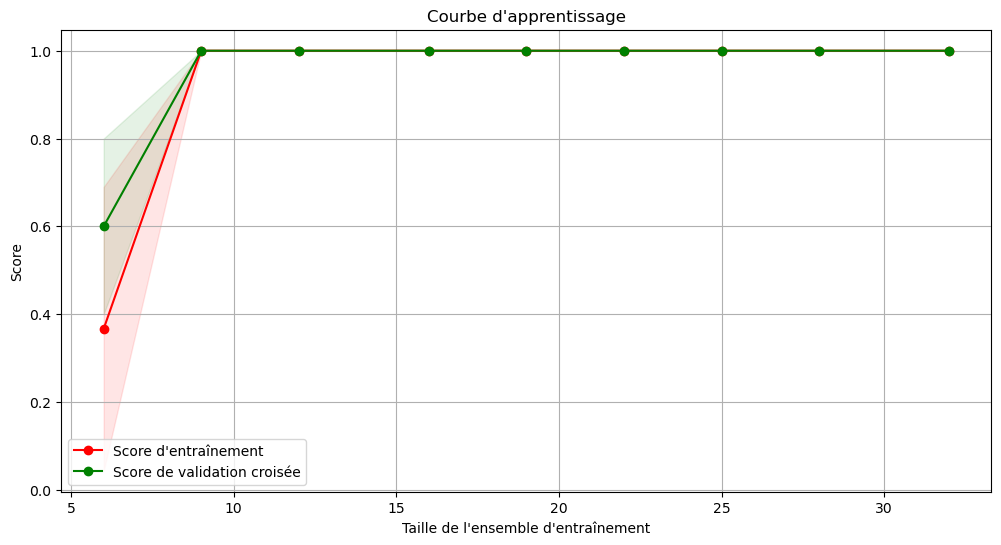
## Résultats de classification

Nous obtenons le rapport de classification suivant :



On peut voir que notre modèle distingue parfaitement les 2 classes. C’était un résultat attendu car les 2 types de sons sont très facilement distinguables.

## Analyse et perspectives



En observant la courbe d’apprentissage, on peut voir que le modèle apprend et généralise très vite. Il n’est pas nécessaire d’avoir une taille d'échantillon d'entraînement plus grande que 10, car le modèle atteint déjà 100% de bonne prédiction à ce stade.

Pour ce qui est des perspectives intéressantes à explorer, on pourrait chercher à ajouter du silence que l’on qualifierait de bruit blanc, pour se préparer à l’eventualité que notre modèle soit confronté à ce cas de figure, et évitre qu’il ne classe le silence comme un sinus.

# Classification de signaux acoustiques réels

## Cadre de la classification

Nous avions des données de sons, un csv et un code d’ouverture des données qui ont été récupérés dans un git hub: <https://github.com/ankitshah009/Task-4-Large-scale-weakly-supervised-sound-event-detection-for-smart-cars>.

Les données étaient mélangées, il y avait des bruits de voitures, camions, trains, radio, sirènes, etc….

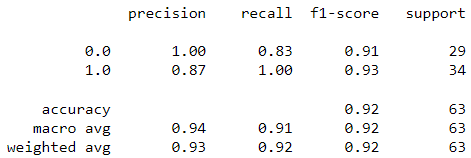
La décision a été prise de nettoyer les données, puis de ne récupérer que celles dont le son en continu concernait les voitures et les camions. Pour y arriver, l’écoute des sons a permis de pouvoir choisir 10 sons de chaques, avec exclusivement des sons de voitures pour les sons voitures et des sons camions pour les sons camions.

Les sons ont ensuite été coupés pour permettre d’avoir plus d’échantillons de chaque.

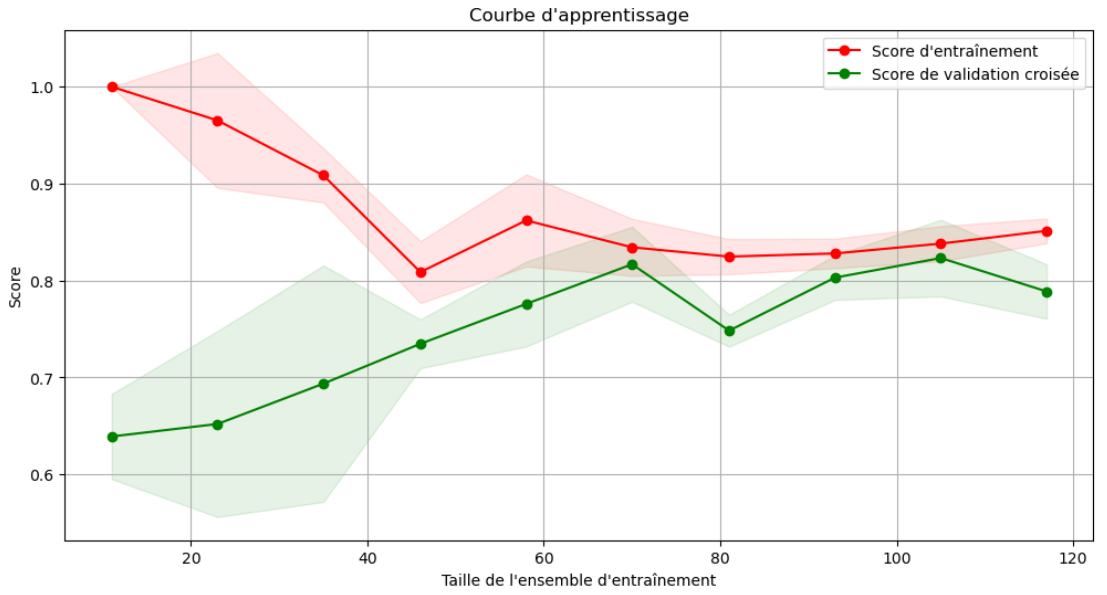
Chaque son a été coupé en 10 échantillons.

## Résultats bruts avec le modèle précédent

Ces échantillons ont été testé sur le modèle créé précédemment, les résultats sont:



## Analyses et pistes d’amélioration

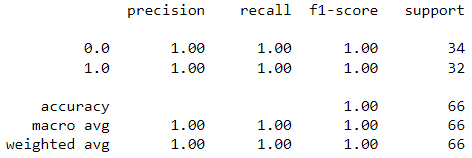


Sur ce graphique, la courbe (Score de validation croisé) augmente progressivement et rejoint sans l’atteindre la courbe (Score d'entraînement).

D'après ces résultats, il a été décidé de découper en plus petit échantillon pour avoir plus de données dans notre base.

## Résultats améliorés

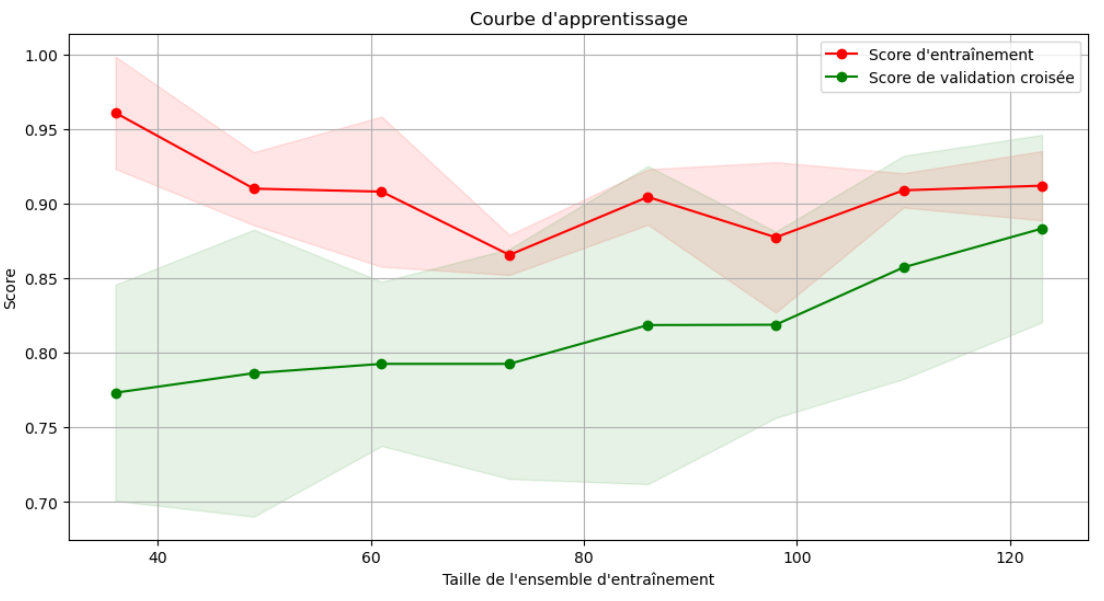
Dans cette partie, les sons ont été divisés en 20, voici le rapport de classification:



Dans ce rapport, l’accuracy semble parfaite, mais il nous faudrait tout de même observer le score de validation croisée pour savoir si notre modèle arrive à bien généraliser.

## Analyse et perspectives supplémentaires

Ci-dessous la learning curve:

****

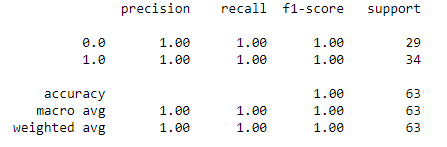
On voit que nous avons un modèle qui apprend bien. Il serait intéressant de fournir encore un peu plus de données au modèle, car nous n’avons pas atteint un seuil stable sur cette courbe d’apprentissage. Il est possible que notre modèle atteigne les 90% de précision.

# Mise en place des méthodes ensemblistes

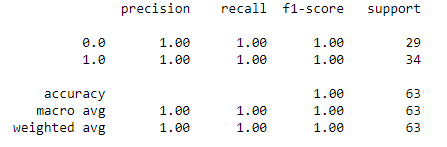
La mise en place de méthodes ensemblistes pourrait nous permettre de gagner en performance. Nous allons donc tenter 3 approches différentes : Le bagging, le boosting et le stacking

## Résultats avec les méthodes ensemblistes

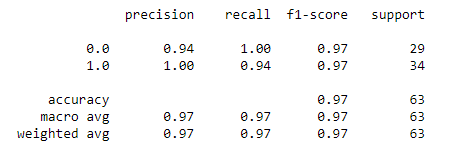
Bagging :



Boosting :



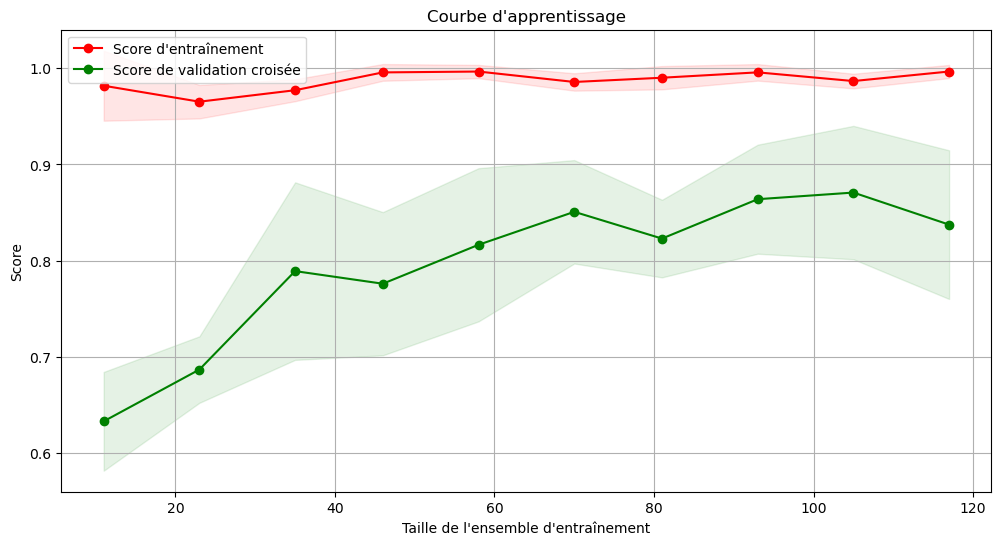
Stacking :



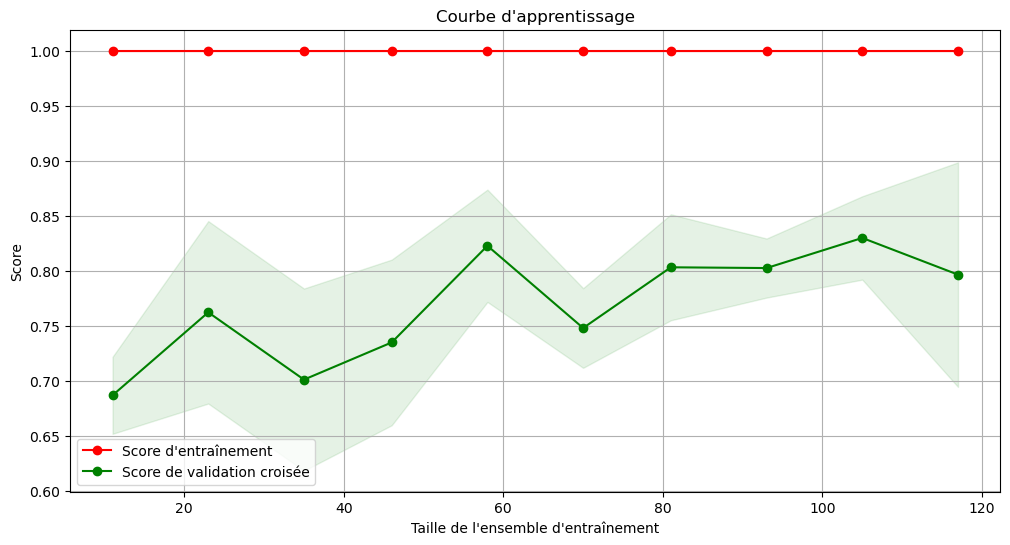
Comme on peut le voir, nous avons effectivement gagné en performance sur les données d’entraînement, pour atteindre le 100% de prédictions correctes pour le boosting et le bagging. Nous devrions tout de même analyser les résultats en validation croisée pour savoir si ces modèles arrivent à correctement généraliser.

## Analyse des résultats

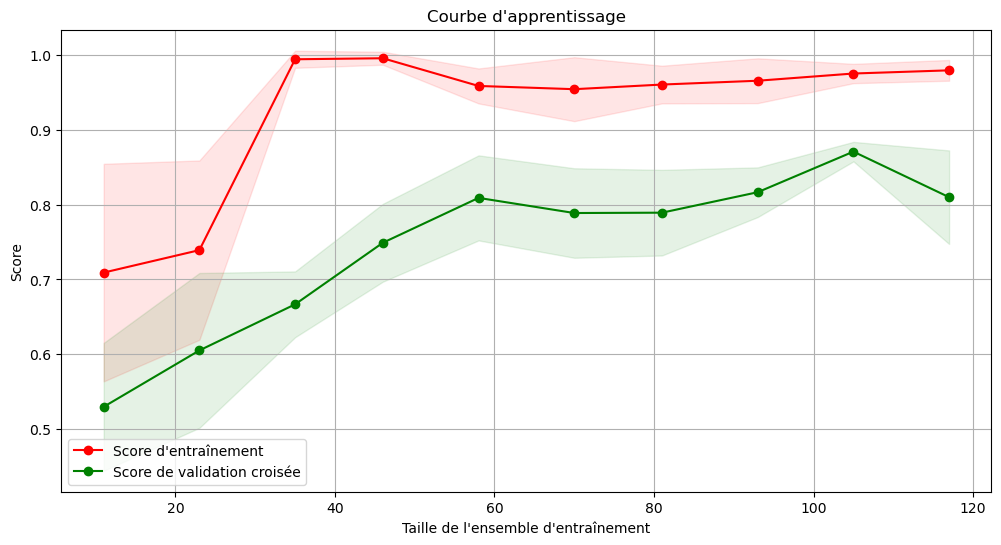
Bagging :



On peut voir que le modèle apprend très correctement, même s’il overfit tout de même légèrement

Boosting :

Même conclusion, même si les performance sont un peu moindre

Stacking :

Même conclusion à tirer pour le stacking. Dans ce contexte, il semble que le bagging soit la solution la plus adaptée.

## Améliorations à apporter

# En termes d’améliorations, nous pourrions tenter de faire varier différents hyperparamètres pour observer l’évolution des performances. De plus, nous pourrions observer le temps d'exécution de chaque modèle, en fonction de la taille des échantillons d'entraînement. Nous pourrions ainsi voir si certains modèles que nous avons plus efficaces comme nous avions jugés plus efficaces ne sont pas trop gourmands par rapport au gain de précision.

Une autre piste intéressante à explorer serait de chercher à distinguer plus de types de sons différents, que ce soit de la parole, des trains… Nous pourrions ainsi voir à partir de quel nombre de classes quels modèles deviennent plus pertinents.

# Conclusion

Suite aux différentes analyses effectuées dans ce rapport, voici nos recommandations d’utilisation des caractéristiques acoustiques et des méthodes pour la reconnaissance sonore de signaux acoustiques.