

# MINI-PROJET : CALCUL SCIENTIFIQUE ET APPRENTISSAGE AUTOMATIQUE

Mohamed amine KASMI - Hiba L'MOUDDEN

29 October 2023

## 1 Introduction

La beatbox, une forme d'expression artistique qui consiste à produire des sons de batterie et de percussion avec la bouche, est devenue un domaine fascinant de la musique contemporaine. Elle implique des techniques vocales et articulatoires avancées pour imiter les sons de la batterie de manière réaliste. La beatbox offre un potentiel d'exploration considérable dans le domaine de l'acoustique et de la reconnaissance de motifs sonores.

L'objectif principal de ce travail pratique (TP) est de développer un système de détection et de reconnaissance des différents sons de beatbox à partir de fichiers sonores au format .wav. Notre approche repose sur l'analyse cepstrale, une technique qui permet de représenter les caractéristiques spectrales d'un signal audio. Grâce à cette analyse, nous cherchons à identifier et à classer les 12 sons de beatbox.

La réalisation de cette tâche est un défi passionnant qui combine des concepts de traitement du signal, d'analyse spectrale et de classification. La précision dans la reconnaissance des sons de beatbox peut avoir des applications significatives dans la musique, la production audio, l'industrie du divertissement et la recherche en acoustique.

ce projet allie la passion pour la musique, la science de l'acoustique et les avancées technologiques pour créer un système de reconnaissance innovant pour les sons de beatbox. La réussite de ce TP ouvrira de nouvelles perspectives pour l'application de l'analyse cepstrale dans la reconnaissance de motifs sonores, tout en contribuant à la compréhension et à la préservation de cette forme d'expression artistique unique.

## 2 Classification des sons beatbox sans prétraitement

Dans cette première partie du projet, l'objectif principal était de classer des sons beatbox sans effectuer de prétraitement préalable. Pour atteindre cet objectif, deux approches ont été explorées : une méthode d'apprentissage supervisé et une méthode d'apprentissage non supervisé.

Dans cette première partie du projet, l'objectif principal était de classer des sons beatbox sans effectuer de prétraitement préalable. Pour atteindre cet objectif, deux approches ont été explorées : une méthode d'apprentissage supervisé et une méthode d'apprentissage non supervisé.

Dans le cadre de l'approche supervisée, une méthode spécifique a été choisie, et les sons beatbox ont été classifiés en utilisant soit les codes des travaux pratiques précédents, soit la puissante bibliothèque scikit-learn. Cette méthode supervisée a permis d'obtenir des résultats classifiés. Pour évaluer la performance de cette classification, une matrice de confusion a été calculée, ainsi qu'un score de performance (*accuracy\_score*). Ces mesures nous ont fourni des informations cruciales sur la capacité du modèle à distinguer les différents types de sons beatbox, et l'image des signaux présentée ci-dessus a été utilisée pour interpréter les résultats.

## 2.1 Méthode supervisée :

Dans ce premier extrait de code, nous avons utilisé l'algorithme K-Nearest Neighbors (KNN) pour effectuer la classification des données. L'objectif de cette section est de classer les données en utilisant un modèle KNN avec 10 voisins les plus proches.

Tout d'abord, nous avons créé une instance de la classe `KNeighborsClassifier`. Ensuite, nous avons ajusté (entraîné) le modèle KNN en utilisant les données d'entraînement avec les caractéristiques `X_train` et les étiquettes de classe correspondantes `y_train`. Après l'apprentissage du modèle, nous l'avons utilisé pour effectuer des prédictions sur un ensemble de données de test, et les prédictions ont été stockées dans la variable `y_pred`, pour évaluer la performance du modèle, nous avons affiché les prédictions en utilisant `print(y_pred)`.

```
knn = KNeighborsClassifier(n_neighbors=10)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
print(y_pred)
```

Figure 1:

Dans ce deuxième extrait de code, nous avons poursuivi l'évaluation de la classification KNN réalisée précédemment. L'objectif est d'évaluer la qualité des prédictions effectuées par le modèle KNN et de calculer des métriques de performance.

Tout d'abord, nous avons choisi le nombre de voisins à utiliser pour la classification, dans ce cas,  $k=10$ .

Ensuite, nous avons défini le nombre de données à tester, qui est égal au nombre de lignes dans l'ensemble de données de test, `Nt=X_test.shape[0]`.

Nous avons ensuite affiché les résultats de prédiction (y pred) et la vérité terrain (y test) pour comparer les prédictions du modèle avec les étiquettes de classe réelles.

Pour évaluer la performance du modèle, nous avons calculé la matrice de confusion en utilisant la fonction confusion matrix, puis nous l'avons affichée. La matrice de confusion nous permet de comprendre combien de prédictions étaient correctes et combien étaient incorrectes.

Enfin, nous avons calculé la précision du modèle en utilisant la fonction accuracy score et avons affiché le résultat. La précision est une mesure importante de la qualité de la classification, car elle indique la proportion de prédictions correctes parmi toutes les prédictions effectuées.

```
# Choix du nombre de voisins
k=10
# Nombre de données à tester
Nt=X_test.shape[0]
#Nt_test = int(Nt/10); # A changer, pouvant aller jusqu'à Nt

# Affichage des résultats de prédiction et de vérité terrain
MatriceConfusion=np.zeros((10,10))
print('Resultat Kppv',y_pred)
print('Vérité terrain',y_test)

# Calcul de la matrice de confusion
confusion = confusion_matrix(y_test, y_pred)

# Affichage de la matrice de confusion
print(confusion)

# Calcul de la précision
accuracy = accuracy_score(y_pred, y_test)

# Affichage de la précision
print("Précision :", accuracy)
```

Figure 2:

## 2.2 Méthode non supervisée :

Dans cette partie de notre projet, nous avons exploré une méthode de classification non supervisée basée sur l'algorithme de K-Means. L'objectif était de regrouper nos données en clusters, sans utiliser d'étiquettes de classe préexistantes. Nous avons choisi d'utiliser l'algorithme K-Means pour cette tâche.

Tout d'abord, nous avons créé un modèle K-Means en spécifiant le nombre de clusters (k) que nous souhaitons obtenir. Dans ce cas, nous avons choisi d'utiliser 12 clusters. Ensuite, nous avons entraîné le modèle K-Means en utilisant les données disponibles (dans la variable data). Le modèle K-Means a cherché à regrouper les points de données en 12 clusters en fonction de leurs similarités. Une fois le modèle K-Means entraîné, nous avons obtenu les étiquettes de cluster attribuées à chaque point de données. Ces étiquettes indiquent à quel cluster chaque point de données appartient, ce qui est essentiel pour comprendre la structure du regroupement. Pour mieux interpréter les résultats, nous avons

également utilisé une fonction correspondance pour associer ces étiquettes de cluster avec des libellés plus explicites. Cette étape a permis de mieux comprendre la signification de chaque cluster en attribuant des noms ou des descriptions aux étiquettes. Les résultats de cette correspondance ont été imprimés pour une meilleure compréhension. Enfin, nous avons évalué la qualité du regroupement en utilisant des métriques telles que la matrice de confusion et le score de performance (accuracy score), qui nous permettent de mesurer à quel point les données ont été bien regroupées dans les clusters. Cela nous aide à comprendre la capacité de l'algorithme K-Means à trouver des structures significatives dans les données, sans l'utilisation de labels de classe préexistants.

```
from sklearn.cluster import KMeans

# Créez un modèle k-means avec un certain nombre de clusters (k)
kmeans = KMeans(n_clusters=12)
# Entraînez le modèle sur vos données
kmeans.fit_predict(data)
# Obtenez les étiquettes de cluster pour chaque point de données
labels = kmeans.labels_

l = correspondance(labels, label)
print(l)
```

Figure 3:

Après avoir appliqué l'algorithme K-Means pour regrouper nos données en clusters, il est essentiel d'évaluer la qualité du clustering obtenu. Cela nous permet de mesurer à quel point les données ont été bien regroupées dans les clusters, même en l'absence de labels de classe préexistants. Pour ce faire, nous avons utilisé deux métriques importantes : la matrice de confusion et la précision.

Tout d'abord, nous avons calculé la matrice de confusion en comparant les étiquettes de cluster attribuées par l'algorithme K-Means (dans la variable `l`) avec les véritables étiquettes (dans la variable `label`). La matrice de confusion est un tableau qui montre le nombre de points de données correctement attribués à chaque cluster ainsi que le nombre de points mal attribués. Cette matrice nous permet de visualiser la qualité du regroupement obtenu. Ensuite, nous avons calculé la précision du clustering en utilisant la métrique accuracy score. La précision mesure la proportion de points de données correctement regroupés dans leurs clusters respectifs. Plus la précision est élevée, plus le regroupement est considéré comme de haute qualité. Ces évaluations nous aident à quantifier la performance de notre algorithme K-Means en termes de clustering et à déterminer si les clusters obtenus sont cohérents et significatifs. Ces mesures sont essentielles pour comprendre à quel point le regroupement est approprié pour notre tâche.

```

confusion = confusion_matrix(label, l)
print(confusion)

accuracy = accuracy_score(l, label)
print("Précision :", accuracy)

```

Figure 4:

### 3 Classification des sons beatbox avec prétraitement

#### 3.1 Méthode supervisée avec réduction de dimension par ACP sur l'ensemble d'apprentissage

Dans cette deuxième partie de notre projet, nous avons exploré une approche de classification supervisée des sons beatbox en utilisant la Réduction de Dimension par l'Analyse en Composantes Principales (ACP) sur l'ensemble d'apprentissage. L'objectif était de réduire la dimension des données tout en préservant l'information essentielle, puis de classer les données réduites à l'aide d'un modèle supervisé de notre choix. Dans cette section, nous expliquons en détail les étapes du processus. Tout d'abord, nous avons appliqué l'Analyse en Composantes Principales (PCA) sur l'ensemble d'apprentissage. L'ACP est une technique de réduction de dimension qui permet de projeter les données originales dans un espace de dimension inférieure tout en maximisant la variance des données. Cela nous permet de représenter les données de manière plus compacte et de faciliter la visualisation. Après avoir appliqué l'ACP, nous avons obtenu de nouvelles données réduites en dimension,  $X_{\text{train pca}}$  pour l'ensemble d'apprentissage et  $X_{\text{test pca}}$  pour l'ensemble de test.

##### Classification Supervisée avec SVM

Pour effectuer la classification supervisée, nous avons choisi d'utiliser un modèle de Support Vector Machine (SVM), qui est couramment utilisé pour la classification. Nous avons créé un modèle SVM et l'avons entraîné sur les données d'apprentissage réduites en dimension. Une fois le modèle SVM entraîné, nous avons utilisé ce dernier pour prédire les étiquettes de classe sur les données de test réduites en dimension. Pour évaluer la performance de la classification supervisée, nous avons calculé la matrice de confusion en comparant les étiquettes prédites ( $y_{\text{pred svm}}$ ) avec les étiquettes de test réelles ( $y_{\text{test}}$ ). La matrice de confusion nous permet de visualiser les prédictions correctes et incorrectes pour chaque classe. Enfin, nous avons calculé le score de performance en utilisant la métrique de l'exactitude (accuracy score). Ce score mesure la proportion de prédictions correctes parmi toutes les prédictions effectuées. Les résultats de la matrice de confusion et de l'exactitude sont essentiels pour évaluer à quel point notre modèle supervisé avec ACP est capable de classer les sons beatbox de manière précise. Ces évaluations nous permettent de mesurer la qualité de notre modèle de classification dans un contexte de réduction de dimension.

```

from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline
# Créez un objet PCA
pca = PCA(n_components=5)

# Entraînez l'ACP sur les données d'entraînement et transformez-les
X_train_pca = pca.fit_transform(X_train)

# Appliquez la transformation ACP aux données de test
X_test_pca = pca.transform(X_test)

```

Figure 5:

```

from sklearn.svm import SVC
# Créez un modèle SVM
svm_classif = SVC()

# Entraînez le modèle SVM sur les données d'apprentissage réduites en dimension
svm_classif.fit(X_train_pca, y_train)

# Prédisez les étiquettes sur les données de test réduites en dimension
y_pred_svm = svm_classif.predict(X_test_pca)

# Calculez la matrice de confusion
confusion = confusion_matrix(y_test, y_pred_svm)

# Calculez le score de performance (exactitude)
accuracy = accuracy_score(y_test, y_pred_svm)

print("Matrice de confusion de la classification supervisée avec ACP :\n", confusion)
print("Score de performance : {:.2f}".format(accuracy))

```

Figure 6:

### 3.2 Méthode non supervisée avec réduction de dimension par ACP sur l'ensemble des données

Dans cette section de notre projet, nous avons exploré une méthode de classification non supervisée des sons beatbox en utilisant l'Analyse en Composantes Principales (ACP) sur l'ensemble des données. L'objectif était de réduire la dimension des données tout en préservant les informations essentielles, puis d'appliquer une méthode non supervisée pour regrouper les données. Voici une explication détaillée de cette approche : Tout d'abord, nous avons appliqué l'Analyse en Composantes Principales (PCA) sur l'ensemble des données. L'ACP est une technique de réduction de dimension qui permet de projeter les données originales dans un espace de dimension inférieure tout en maximisant la variance des données. Cette étape nous a permis de représenter les données de manière plus compacte. Après avoir appliqué l'ACP, nous avons obtenu de nouvelles données réduites en dimension, data\_pca. Nous avons choisi d'utiliser l'algorithme de K-Means pour regrouper les données réduites en dimension en

clusters. Le nombre de clusters (n clusters) a été défini comme égal au nombre de classes de sons de beatbox. K-Means cherche à regrouper les données en fonction de leurs similarités sans utiliser d'étiquettes de classe préexistantes. Une fois le modèle K-Means entraîné, nous avons obtenu les clusters prédits (y pred kmeans). Nous avons ensuite associé ces clusters prédits avec des étiquettes de classe pour mieux comprendre la signification de chaque cluster. Pour évaluer la performance de la classification non supervisée, nous avons calculé la matrice de confusion en comparant les étiquettes réelles (label) avec les clusters prédits (number labels kmeans). La matrice de confusion nous permet de visualiser les prédictions correctes et incorrectes pour chaque classe. Enfin, nous avons calculé le score de performance en utilisant la métrique de l'exactitude (accuracy score). Ce score mesure la proportion de prédictions correctes parmi toutes les prédictions effectuées. Les résultats de la matrice de confusion et de l'exactitude sont essentiels pour évaluer à quel point notre méthode de clustering K-Means sur les données ACP est capable de regrouper les sons beatbox de manière précise. Ces évaluations nous permettent de mesurer la qualité du clustering non supervisé dans un contexte de réduction de dimension.

```
# Créez un modèle K-Means avec le nombre de clusters égal au nombre de classes de sons de beatbox
n_clusters = len(np.unique(label))
kmeans = KMeans(n_clusters=n_clusters, n_init=10, random_state=42)
data_pca = pca.fit_transform(data)
# Entraînez le modèle K-Means sur les données ACP
kmeans.fit(data_pca)

# Obtenez les clusters prédits
y_pred_kmeans = kmeans.predict(data_pca)

number_labels_kmeans = correspondance(y_pred_kmeans, label)
```

Figure 7:

```
# Calculer la matrice de confusion
confusion_kmeans = confusion_matrix(label, number_labels_kmeans)

# Calculer le score de performance (exactitude)
accuracy_kmeans = accuracy_score(label, number_labels_kmeans)

print("Matrice de confusion du K-Means sur les données ACP :\n", confusion_kmeans)
print("Score de performance (exactitude) du K-Means sur les données ACP : {:.2f}".format(accuracy_kmeans))
```

Figure 8:

## 4 Analyse des resultas

### 4.1 methode supervisée

La figure 1 : Représente les prédictions du modèle k-NN pour les données de test. Chaque valeur dans ce tableau est une prédiction de classe pour un échantillon donné dans les données de test . Son resultat est le suivant:





figure 4: la matrice de confusion montre comment le modèle K-means a attribué les échantillons aux clusters, et la précision mesure à quel point ces clusters correspondent aux classes réelles. Une précision élevée indique une bonne correspondance entre les clusters et les classes réelles, ce qui est généralement souhaitable dans le regroupement de données en apprentissage non supervisé. Cependant, il est important de noter que le K-means est un algorithme non supervisé, et la correspondance exacte entre les clusters et les classes réelles n'est pas garantie, car il s'agit d'une tâche de regroupement et non de classification. Le resultat est le suivant :

```
[[24  4  0  0  0  0  0  0  0  0  0  0]
 [ 0 18  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 32  1  0  0  0  1  0  1  2  0]
 [ 1  0  0 18  1  0  0  0  0  0  0  0]
 [ 0  0  0  0 24  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 25  0  0  0 12  0  0]
 [ 0  0  0  0  0  1 41  0  0  0  0  0]
 [ 0  0  0  0  1  0  0 24  0  0  0  0]
 [ 4  0  0  0  0  0  0  0  0 33  0  1]
 [ 0  0  0  0  0  5  1  2  0 22  0  0]
 [ 0  0  0  0  0  0  0  0  0  0 22  0]
 [ 0  0  0  0  0  0  0  1  0  0  0 19]]
Précision : 0.8856304985337243
```

Figure 12:

### 4.3 Méthode supervisée avec réduction de dimension par ACP sur l'ensemble d'apprentissage

La figure 6 : les résultats montrent que le modèle SVM, combiné à une réduction de dimension par ACP, a une performance raisonnable pour la classification des données. L'exactitude de 87 indique que le modèle a une bonne capacité de classification, mais il y a encore quelques erreurs de classification. La matrice de confusion permet d'identifier quelles classes sont plus sujettes à des erreurs de classification. Pour une évaluation plus détaillée, vous pourriez examiner d'autres métriques de performance ou explorer davantage les erreurs spécifiques faites par le modèle. Le resultat est la figure 13.

### 4.4 Méthode non supervisée avec réduction de dimension par ACP sur l'ensemble des données

les résultats montrent que le modèle K-Means, appliqué sur les données ACP, a une performance raisonnable pour le regroupement des données. L'exactitude de 72 pour cent indique que le modèle a réussi à regrouper une grande partie des données correctement, mais il y a encore des erreurs de regroupement, ce

```

Matrice de confusion de la classification supervisée avec ACP :
[[ 7  1  0  0  0  0  0  0  1  0  0  0]
 [ 0  6  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 14  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  2  2  0  0  0  0  0  1  0]
 [ 0  0  0  1 11  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 10  2  0  0  0  0  0]
 [ 0  0  0  0  0  0 12  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  5  0  0  0  1]
 [ 1  0  0  0  0  0  0  0 11  0  0  0]
 [ 0  0  0  0  0  1  0  0  0  5  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  9  0]
 [ 0  0  0  0  0  0  0  2  0  0  0  6]]
Score de performance : 0.87

```

Figure 13:

```

Matrice de confusion du K-Means sur les données ACP :
[[22  4  0  0  0  0  0  0  0  0  2  0]
 [ 0 18  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 36  0  1  0  0  0  0  0  0  0]
 [ 0  0  1  0 18  0  0  1  0  0  0  0]
 [ 0  0  0  0 24  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 18 19  0  0  0  0  0]
 [ 0  0  0  0  0  0 42  0  0  0  0  0]
 [ 0  3  3  0  0  0  0 16  0  0  2  1]
 [ 7  0  0  0  0  0  0  0 31  0  0  0]
 [ 5  0  0  0  0  1 22  0  0  0  2  0]
 [ 2  0  0  0  0  0  0  0  0  0 20  0]
 [ 0  0  0  0  0  0  0  2  0  0  0 18]]
Score de performance (exactitude) du K-Means sur les données ACP : 0.72

```

Figure 14:

qui est attendu car K-Means est un algorithme non supervisé et ne connaît pas les vraies classes des données. La matrice de confusion peut être utilisée pour examiner plus en détail les erreurs de regroupement spécifiques

## 5 Conclusion

En conclusion, ces analyses démontrent l'utilisation de différentes méthodes d'apprentissage supervisé (KNN, SVM) et non supervisé (K-Means) en combinaison avec des techniques de réduction de dimension (ACP) pour explorer et analyser des données. Chacune de ces méthodes a ses propres avantages et limitations, et le choix de la méthode dépendra des objectifs spécifiques de votre tâche. Les résultats montrent que ces modèles ont réussi à traiter les données, mais il reste de la place pour l'amélioration, en particulier en termes de réduction des erreurs de classification ou de regroupement.