

# Recommandation

Henri CHABERT, Clara SAMUEL, Coline THERIAL

# Plan

1. Contexte - Objectif
2. Etat de l'art
3. Contribution
  - a. Premiers pas
  - b. Complexité temporelle
  - c. Optimisation du score
  - d. Machine Learning
  - e. Conclusion
4. Nombre de musiques & objectivité
5. Vers une approche visuelle
6. Conclusion

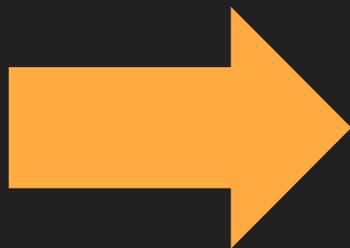
# Contexte - Objectif

Projet réalisé pour  qobuz

- Eviter les **changements** brutaux de musiques dans une **playlist**
- Utiliser directement les **spectrogrammes**

 Objectif retenu :

2  
Musiques



Chainable  
ou  
Non chainable

# Plan

1. Contexte - Objectif
2. Etat de l'art
3. Contribution
  - a. Premiers pas
  - b. Complexité temporelle
  - c. Optimisation du score
  - d. Machine Learning
  - e. Conclusion
4. Nombre de musiques & objectivité
5. Vers une approche visuelle
6. Conclusion

# Articles - Features - Audio - Extraction

Features retenues :

- Zero cross rate
- Sharpness
- Spectral rolloff
- Delta spectrum mag.
- RMS level
- Bandwidth
- Band energy ratio
- MFCC (20)
- Spectral centroid
- Pitch
- Roughness
- Roughness Std. Dev.
- Chroma Stif
- Pitch strength
- BPM



Articles : [Features for Audio and Music Classification](#)

[A large set of features for sound description](#)

[Music feature extraction in Python](#)



Librosa : [Librosa - docs](#)

# Articles - Features - Audio - Extraction

La **bibliothèque Librosa** nous permet d'extraire quasiment toutes les features énoncées plus tôt

En entrée, on passe directement un **fichier audio**, converti ensuite en spectrogramme avec la **fonction load()**



Articles : [Features for Audio and Music Classification](#)

[A large set of features for sound description](#)

[Music feature extraction in Python](#)



Librosa : [Librosa - docs](#)

# Article : Comparaison non supervisée d'albums - Visuel

[Computer analysis of similarities between albums in popular music](#)

## Data Set (**Albums**) :

- Queen (14)
- U2 (11)
- The Beatles (13)
- Tears for fears (6)
- ABBA (8)

## Forme des spectrogrammes :

Fichiers **FLAC** (Free Lossless Audio Codec)

→ Convertis en fichiers **mono WAV**

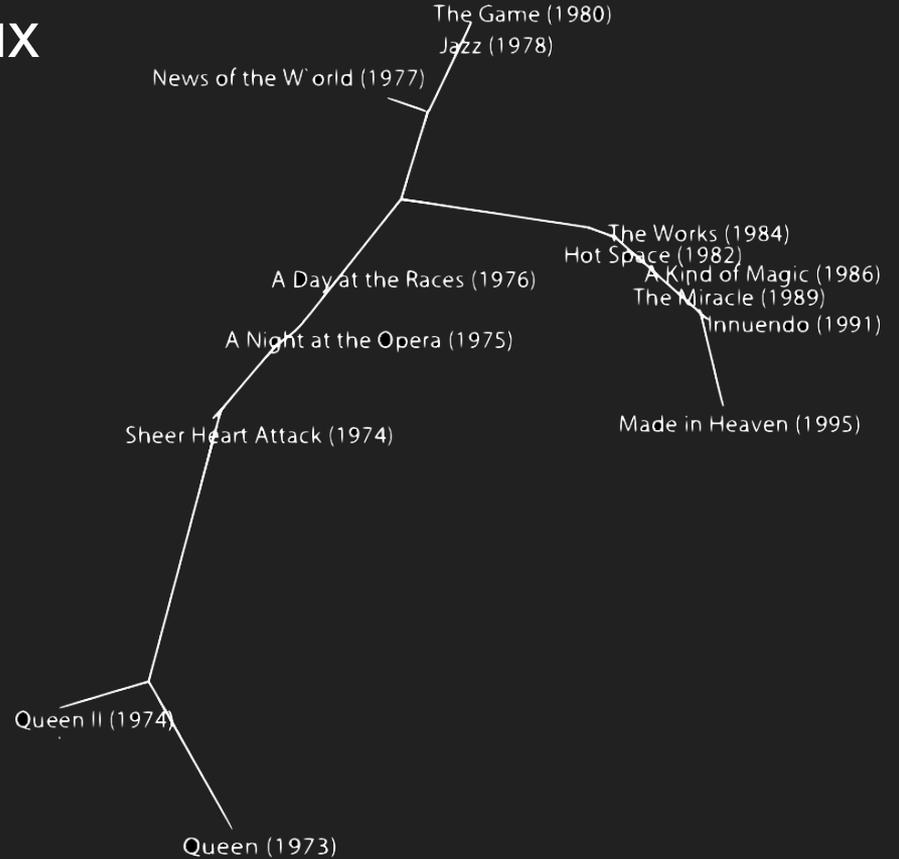
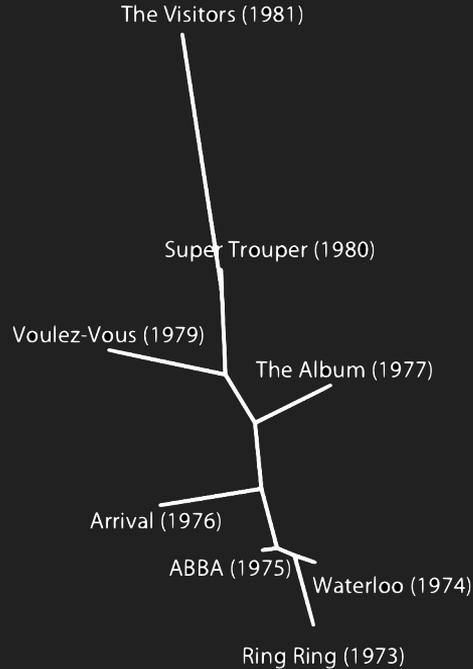
→ Convertis en Spectrogrammes **2D 800x512**  
(fréquence en fonction du temps)

**Echantillon utilisé** : on utilise une partie de la musique seulement, de 00:30 à 1:30

**Features utilisées** : **Wndchrm** feature set (**2883 features**) - **Visuelles**

# Article : Résultats expérimentaux

Représentation par **arbre phylogénique**



# Article : Comparaison non supervisée d'albums - Visuel - Conclusions

## Avantages et inconvénients :

- 😊 Calcul d'un score discriminant pour **pondérer les features**
- 😊 Calcul d'une **distance** entre une chanson et un album
- 😐 **Résultats assez concluants** pour séparer/ordonner des albums connus
- 😞 La comparaison se fait **entre des albums** et non des musiques seules

## A retenir :

- Utiliser des **spectrogrammes partiels** (0:30-1:30)
- Mêmes **features** pour traiter un **spectrogramme en 2D** que pour une **image**

# Plan

1. Contexte - Objectif
2. Etat de l'art
3. Contribution
  - a. Premiers pas
  - b. Complexité temporelle
  - c. Optimisation du score
  - d. Machine Learning
  - e. Conclusion
4. Nombre de musiques & objectivité
5. Vers une approche visuelle
6. Conclusion

# Contribution : 1ère Expérience



Musiques de **Youtube** (téléchargées et sauvegardées avec les informations)



**Extraction des Features** (27 - musicales)



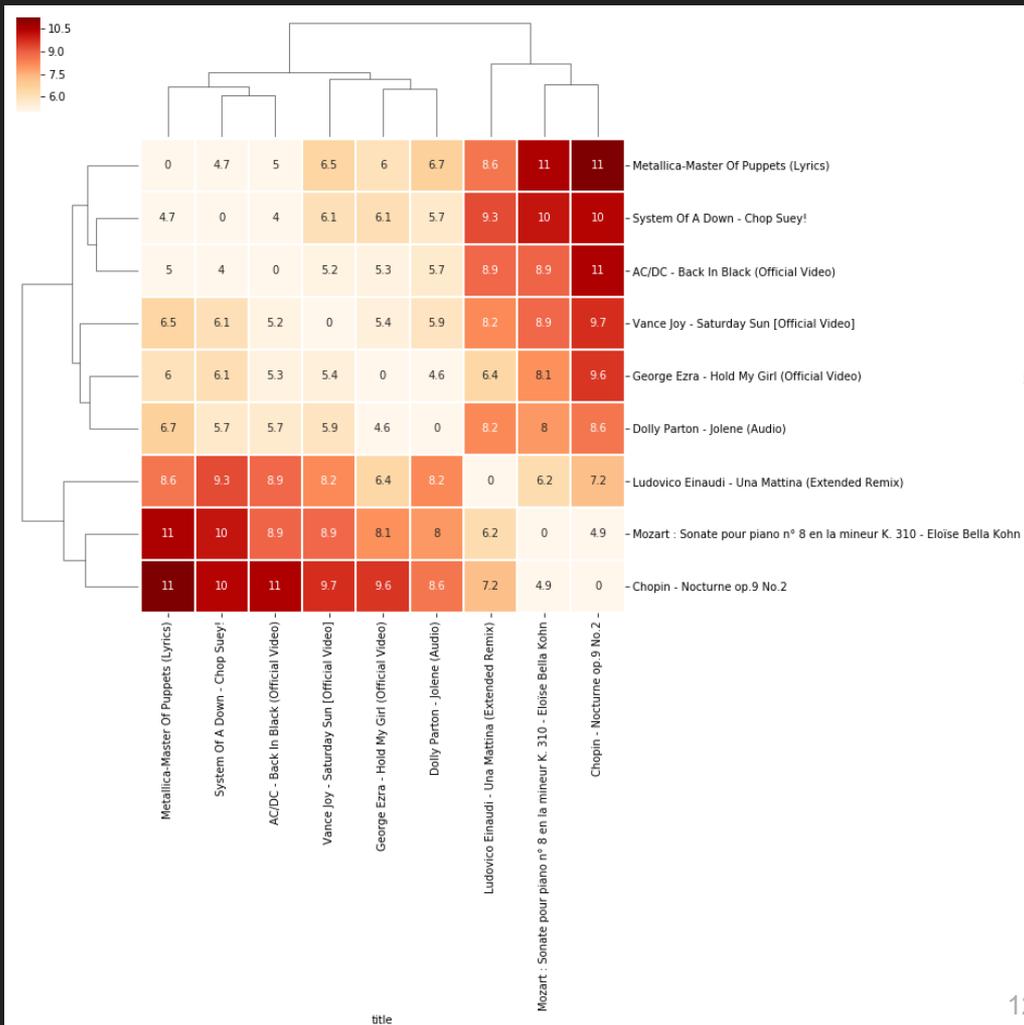
**Distance Euclidienne** 2 à 2 des musiques

# Contribution : Résultats obtenus

↙ ↘  
Peu de différences entre  
les musiques d'un genre

↖ ↗  
Grande différence entre  
Classique et Métal

 Classification



# Contribution : 2ème Expérience



Objectif : Est-ce que 2 musiques sont chaînables ?



Seuils min et max trouvés : 3.6 et 6.8



300 couples de musiques étiquetés (50% chainable)



30s prises en compte entre la 00:30 et 01:00

# Contribution : Analyse des résultats obtenus



65% de réussite



45 Faux négatifs (15%)

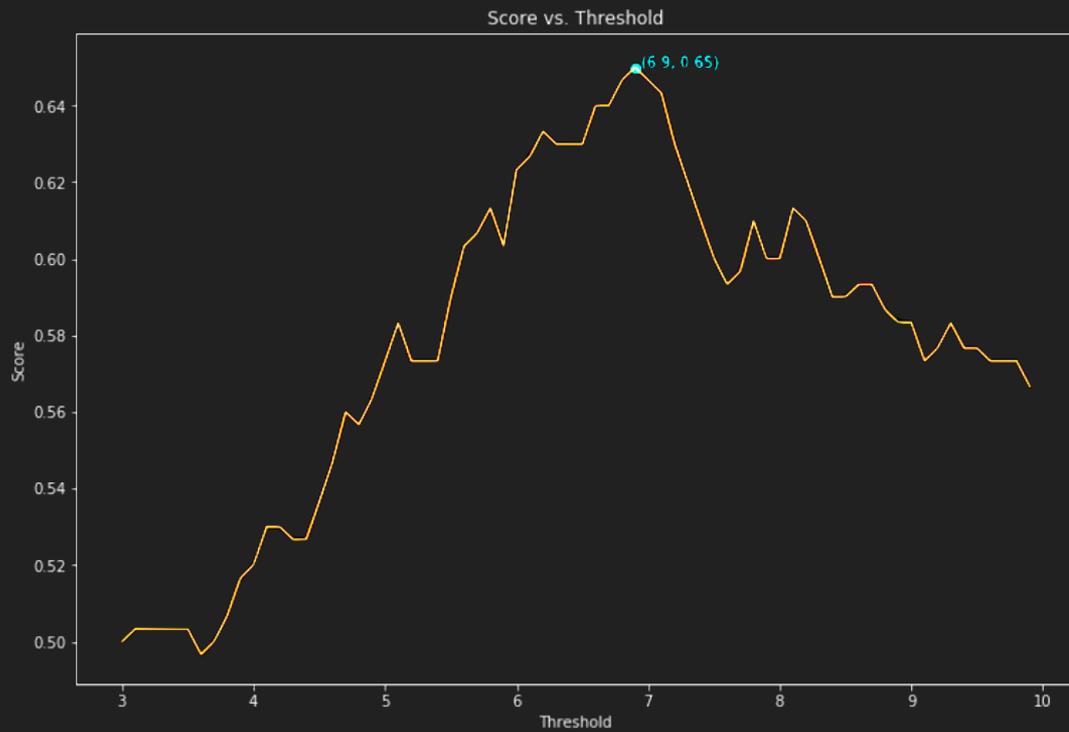


60 Faux positifs (20%)

Red Hots - Californication / U-Turn - Lili: 6.03 de distance → Chainable

Nocturne - Chopin/ Raining Blood : 12.01 de distance → Non chainable

# Contribution : Focus sur le seuil



# Plan

1. Contexte - Objectif
2. Etat de l'art
- 3. Contribution**
  - a. Premiers pas
  - b. Complexité temporelle**
  - c. Optimisation du score
  - d. Machine Learning
  - e. Conclusion
4. Nombre de musiques & objectivité
5. Vers une approche visuelle
6. Conclusion

# Temps - Extraction

Notre **expérience** :

- 30 secondes (0:30 - 1:00)
- 10 itérations → **moyenne**
- Extraction de N musiques

**Résultat** pour N musiques :

- N = 10 → 5.5 secondes
- N = 50 → 31 secondes
- N = 100 → 57 secondes



Le temps d'extraction est **linéaire** par rapport au nombre de musiques



Pour 40 Millions de musiques, l'extraction prendra : 277 jours



# Temps - Calcul des distances (euclidienne)

Notre **expérience** :

- 30 secondes (0:30 - 1:00)
- 10 itérations → **moyenne**
- Calcul de distance pour N paires

**Résultat** pour N paires :

- $N = 10 \rightarrow 0.03$  secondes
- $N = 50 \rightarrow 0.06$  secondes
- $N = 100 \rightarrow 0.13$  secondes



Le temps de calcul de distance est linéaire par rapport au nombre de paires

# Plan

1. Contexte - Objectif
2. Etat de l'art
3. Contribution
  - a. Premiers pas
  - b. Complexité temporelle
  - c. Optimisation du score
  - d. Machine Learning
  - e. Conclusion
4. Nombre de musiques & objectivité
5. Vers une approche visuelle
6. Conclusion

# Score - Secondes pris en compte

**Résultat** pour le temps suivant pris en compte :

- 0:00 - 0:30 → 0.63
- 0:30 - 1:00 → 0.66
- 1:00 - 1:30 → 0.63
  
- 0:30 - 1:30 → 0.66



Pas beaucoup de changement

# Score - Différentes features - Pondération

Notre **expérience** :

- Calculer le score avec **une seule** feature
- Les **comparer**

**Résultats** :

- Sans → 0.65 😞
- Avec → 0.71 😊

**Résultat** pour les features suivantes :

- bpm → 0.53 😞
- chroma\_stft → 0.70 😊
- rms → 0.68
- spectral\_centroid → 0.70
- spectral\_bandwidth → 0.66
- rolloff → 0.70
- zero\_crossing\_rate → 0.70 😊
- mfcc → 0.58

# Score - Distances

**Résultat** avec des distances différentes :

- Euclidienne  $\rightarrow 0.66$
- Manhattan  $\rightarrow 0.68$  😊
- Minkovski ( $p = 4$ )  $\rightarrow 0.60$  😞

**Remarque**, les threshold sont différents en fonction des distances :

- Euclidienne  $\rightarrow 3.5 - 8$
- Manhattan  $\rightarrow 14 - 32$
- Minkovski  $\rightarrow 2 - 5$

# Score Optimisé

Notre **expérience** :

- Secondes : 0:30 - 1:00
- Avec pondération
- Distance Manhattan

**Résultat** :



68%

# Plan

1. Contexte - Objectif
2. Etat de l'art
- 3. Contribution**
  - a. Premiers pas
  - b. Complexité temporelle
  - c. Optimisation du score
  - d. Machine Learning**
  - e. Conclusion
4. Nombre de musiques & objectivité
5. Vers une approche visuelle
6. Conclusion

# Réseau de neurone

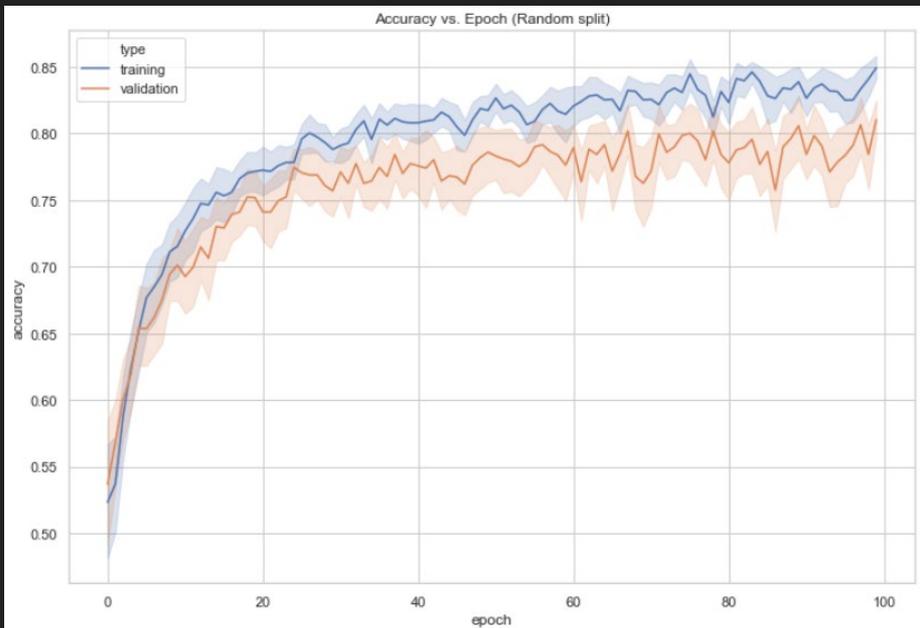
```
def build_model():  
    model = tf.keras.models.Sequential([  
        tf.keras.layers.Flatten(),  
        tf.keras.layers.Dense(27, activation='relu'),  
        tf.keras.layers.Dense(2, activation='softmax')  
    ])  
  
    model.compile(optimizer='adam',  
                  loss='sparse_categorical_crossentropy',  
                  metrics=['accuracy'])  
  
    return model
```

Model: "sequential\_3"

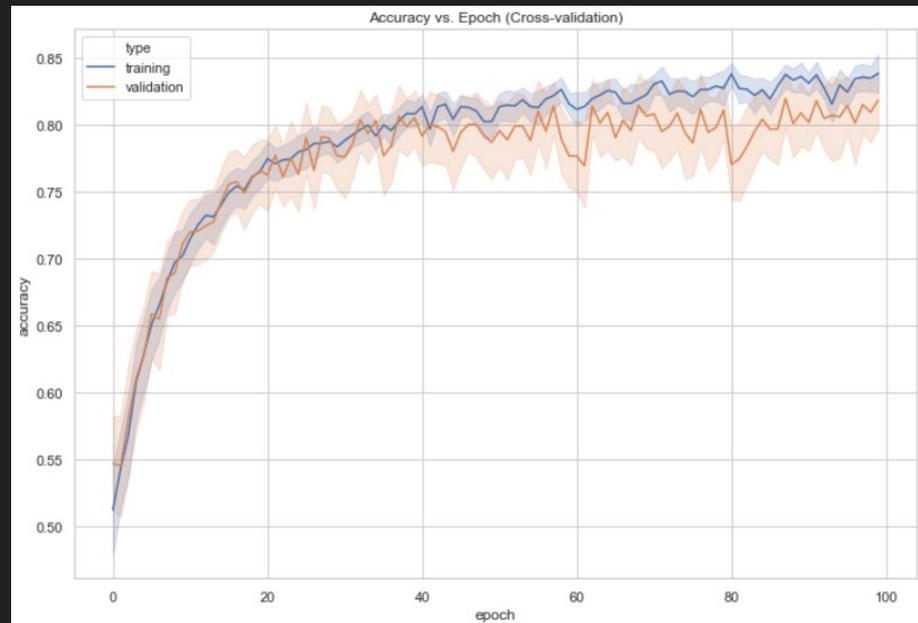
Layer (type)	Output Shape	Param #
flatten_3 (Flatten)	multiple	0
dense_6 (Dense)	multiple	1485
dense_7 (Dense)	multiple	56

=====  
Total params: 1,541  
Trainable params: 1,541  
Non-trainable params: 0  
=====

# Résultat : Réseau de neurone



Random Split



Cross-Validation

# Decision Tree

	precision	recall	f1-score	support
0	0.84	0.93	0.88	56
1	0.85	0.69	0.76	32
accuracy			0.84	88
macro avg	0.84	0.81	0.82	88
weighted avg	0.84	0.84	0.84	88

 Vrai Négatif : 52       Faux Positif : 4

 Faux Négatif : 10       Vrai Positif : 22

# Plan

1. Contexte - Objectif
2. Etat de l'art
3. Contribution
  - a. Premiers pas
  - b. Complexité temporelle
  - c. Optimisation du score
  - d. Machine Learning
  - e. Conclusion
4. Nombre de musiques & objectivité
5. Vers une approche visuelle
6. Conclusion

# Contribution - Conclusion

Contribution	Score
Basique	65%
Basique (avec 00:30-1:00)	65%
Basique avec Pondération	71%
Basique mais Distance Manhattan	66%
Optimisé	68%
Réseau de Neurone (~75 epoch)	~80%
Decision Tree	~84%

# Plan

1. Contexte - Objectif
2. Etat de l'art
3. Contribution
  - a. Premiers pas
  - b. Complexité temporelle
  - c. Optimisation du score
  - d. Machine Learning
  - e. Conclusion
4. **Nombre de musiques & objectivité**
5. Vers une approche visuelle
6. Conclusion

# Nombre de musiques & objectivité



Étiqueté à la main long et fastidieu



Problème d'objectivité

# Ajout de musiques

CEI Lusion

Conan Gray - Comfort Crowd (Live / ...)

vevo DSCVR

CONAN GRAY  
'COMFORT CROWD' LIVE

Sasha Sloan - Too Sad To Cry (Live P...)

SASHA SLOAN TOO SAD TO CRY

vevo LIVE PERFORMANCE



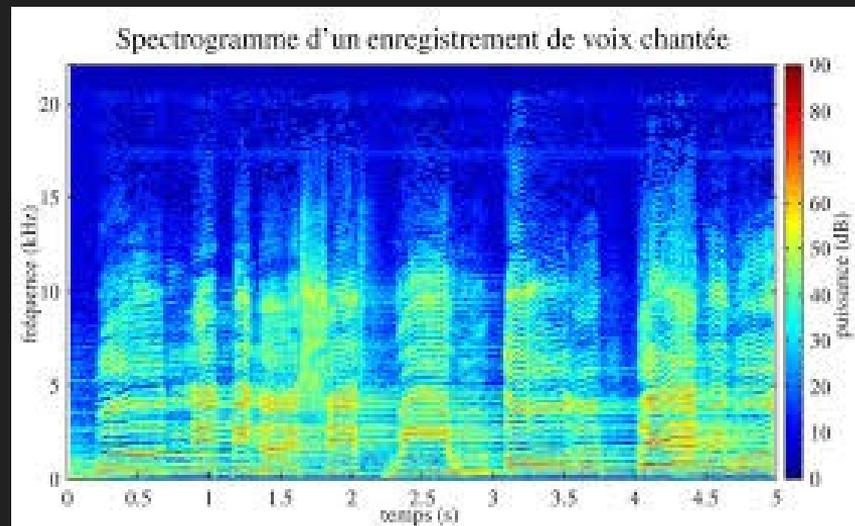
**Idée** : création d'une **interface** Web pour les créer des paires par **crowdsourcing** - **Site**

# Plan

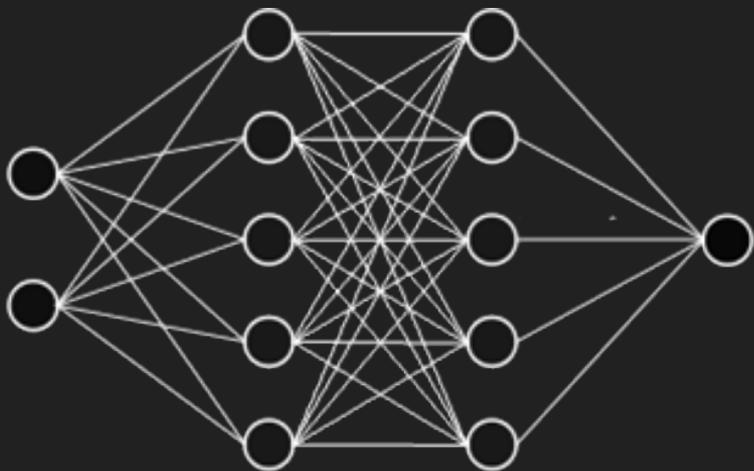
1. Contexte - Objectif
2. Etat de l'art
3. Contribution
  - a. Premiers pas
  - b. Complexité temporelle
  - c. Optimisation du score
  - d. Machine Learning
  - e. Conclusion
4. Nombre de musiques & objectivité
5. **Vers une approche visuelle**
6. Conclusion

# Vers une approche visuelle - Idée

- Features **visuelles**
- Spectrogramme vu comme une **image**
- Techniques de **traitement d'image**

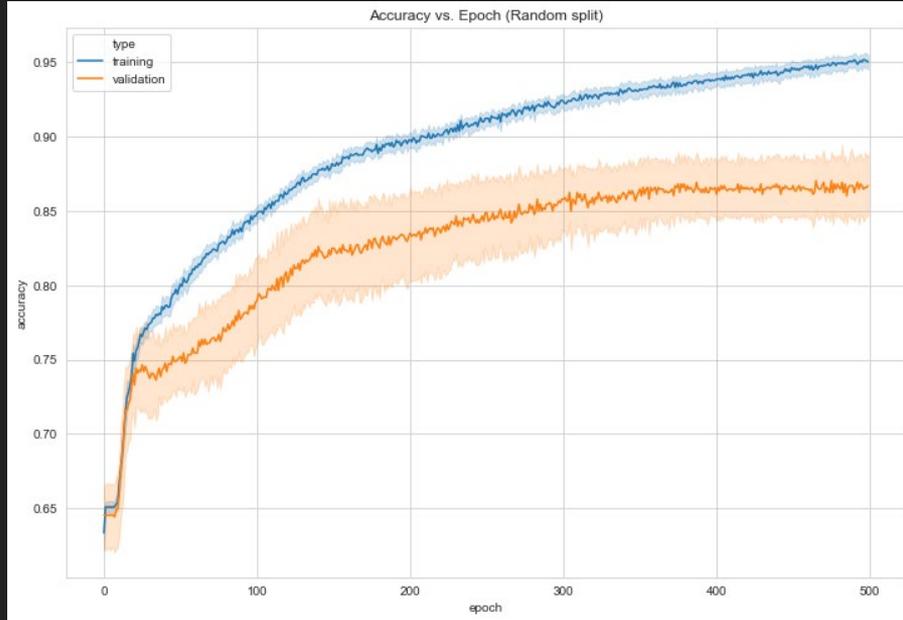


# Vers une approche visuelle - Objectif

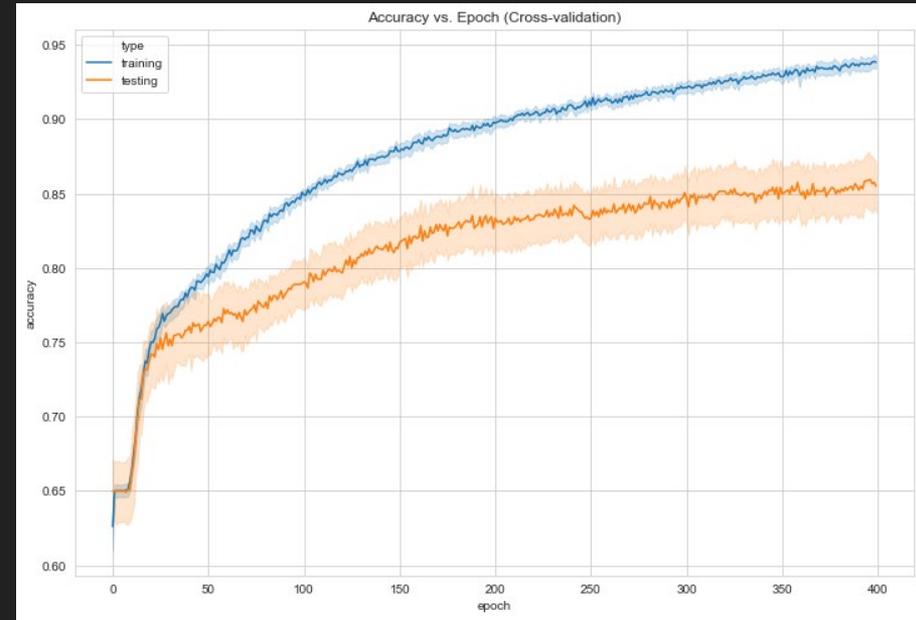


- Extraire des features **visuelles** par CNN
- Les passer dans **notre** Réseau de neurone
- **Comparer** les scores avec les features **audios**
- Comparer avec la **juxtaposition** des deux

# Vers une approche visuelle - Résultat



Random Split



Cross-Validation

# Plan

1. Contexte - Objectif
2. Etat de l'art
3. Contribution
  - a. Premiers pas
  - b. Complexité temporelle
  - c. Optimisation du score
  - d. Machine Learning
  - e. Conclusion
4. Nombre de musiques & objectivité
5. Vers une approche visuelle
6. Conclusion

# Conclusion - Apport pour Lusion

- **Recherche** de différentes méthodes d'analyse de spectres
- **Optimisation** de scores pour chaque méthode retenue
- **Elimination** de pistes potentielles

# Conclusion - Apport personnel

- Introduction au **Machine Learning**
- Utilisation de notebooks **Tensorflow**
- **Recherche** sur les techniques et innovations en matière de **traitement d'image/de son**
- Approfondissement du sujet des **réseaux de neurones**

# Conclusion - Pour aller plus loin...

- **Augmenter l'échantillon étiqueté** de musiques pour le training
- **Mettre à l'échelle** des données de Qobuz
- Augmenter la précision avec un **feedback d'experts** et les **expériences utilisateurs**

# Merci de votre attention

Avez-vous des questions ?