
Projet de parcours



IA



LE CHENADEC Elouarn, ARNAUD Axel

Sommaire.

1

Présentation

2

Défis

3

Architecture(s)

4

Entrainement

5

Résultats

6

Retour sur le projet

Présentation du challenge



Service de gestion de
catalogue destiné à la
production de bibliothèques
musicales

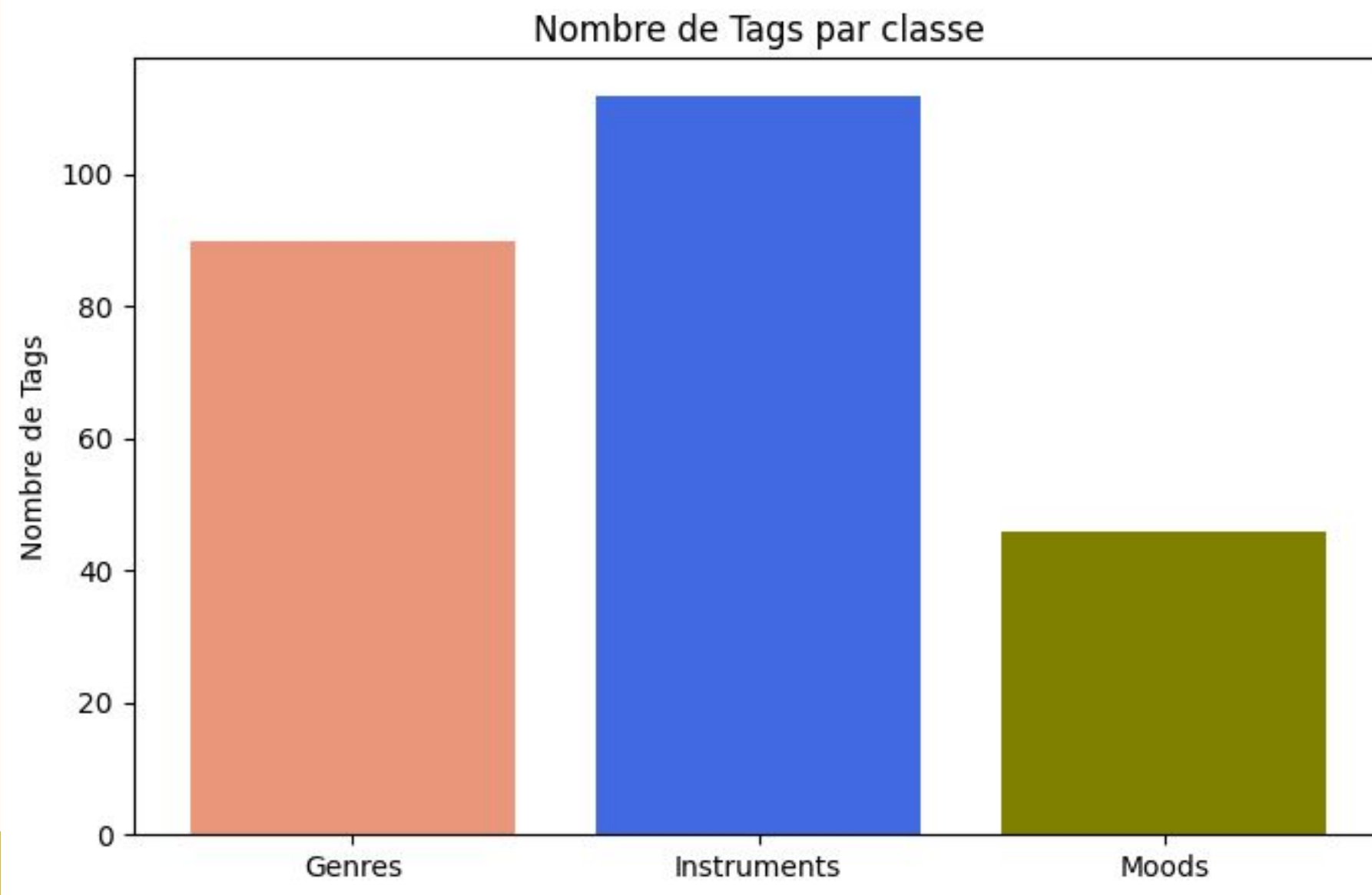
But du challenge :

Optimiser la conversion des prédictions
numériques en décisions concrètes

Battre le benchmark (seuil à 0.5 sur les
probabilités)

Présentation du dataset

1 2 110 000 inputs
3 4 290 features

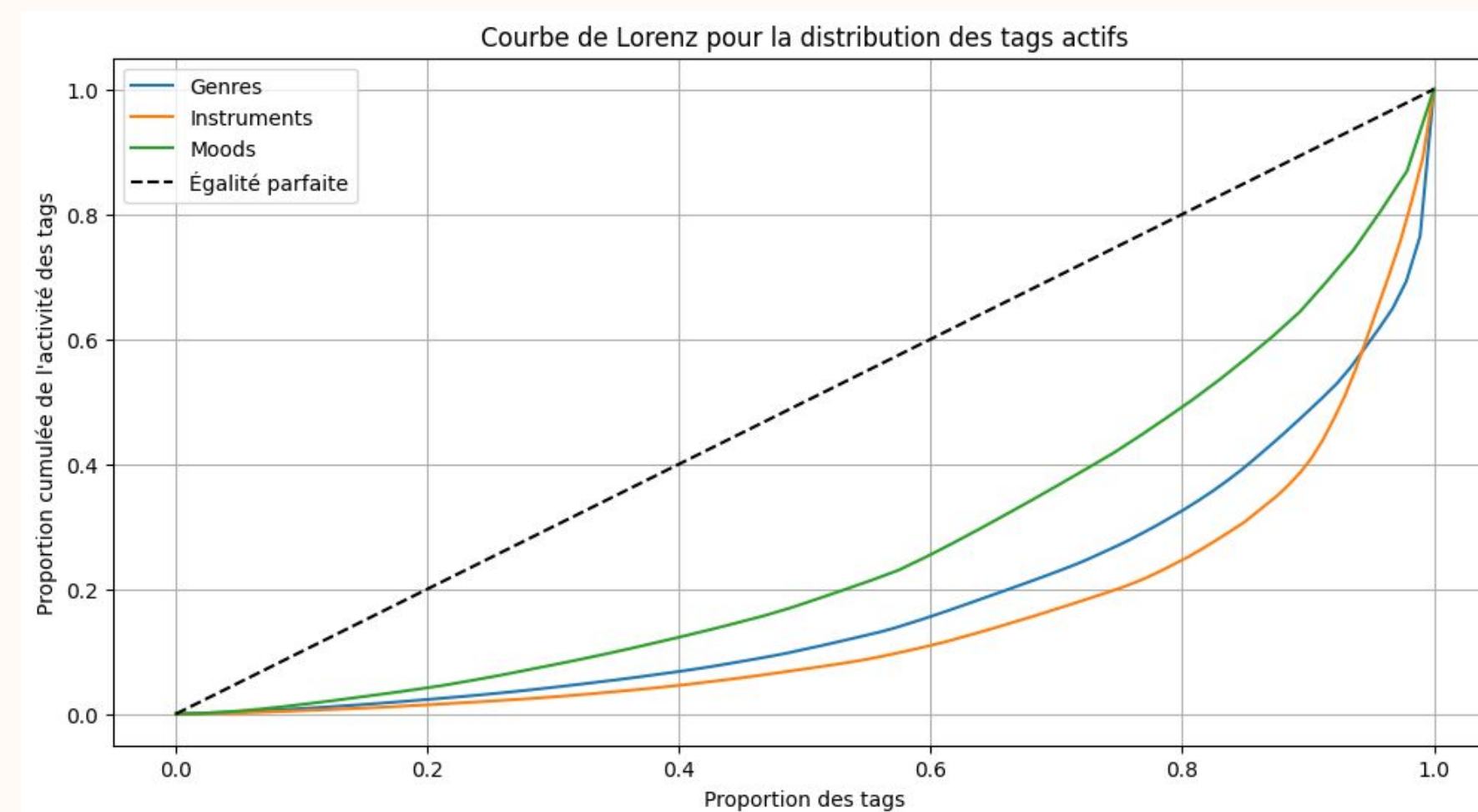
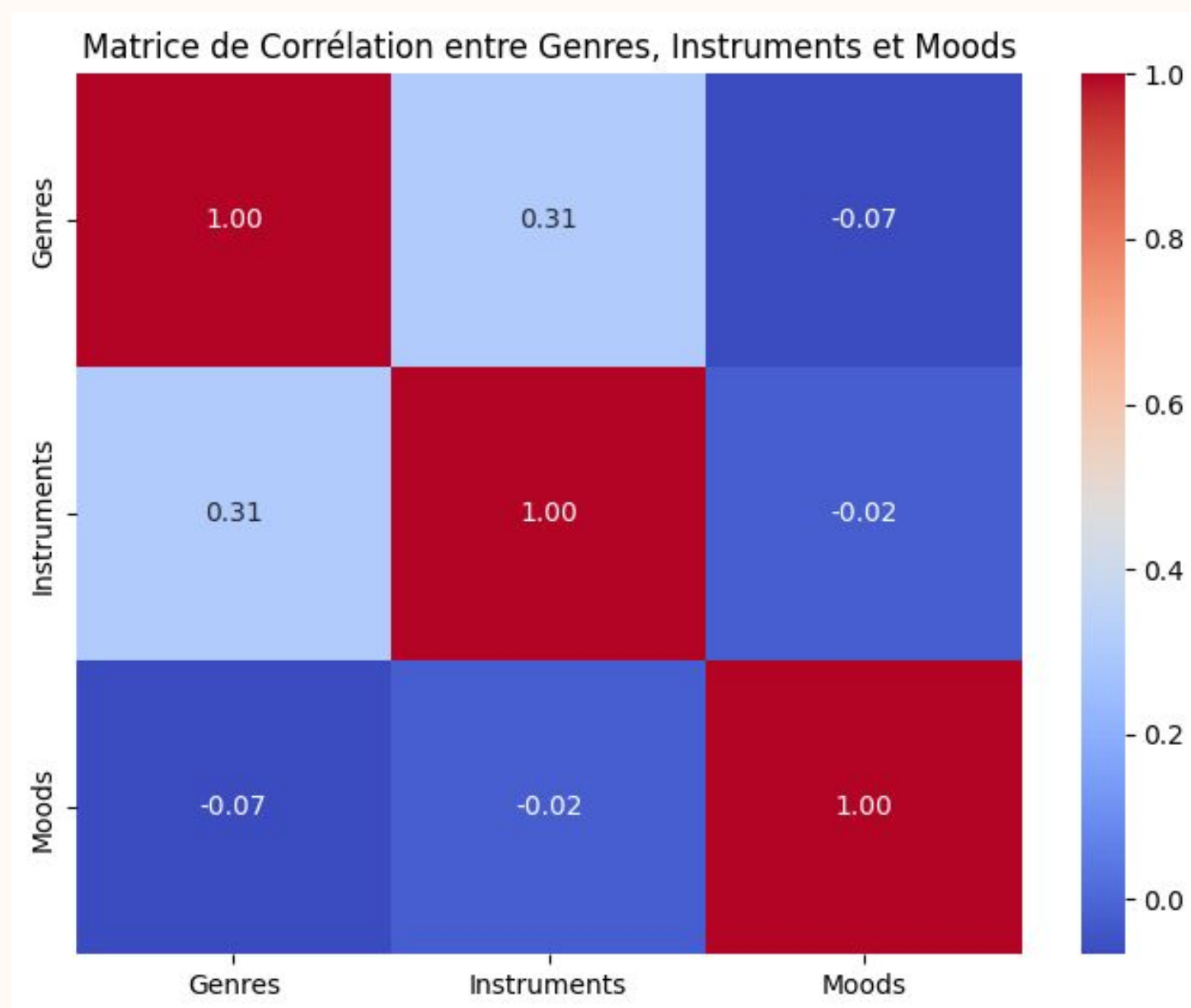


features : probabilités pour un tag de correspondre à la musique (input)

et ~40 tags de catégories

Présentation du dataset

Indice de Gini de
quasi 1 pour chaque
classe



Défis du dataset

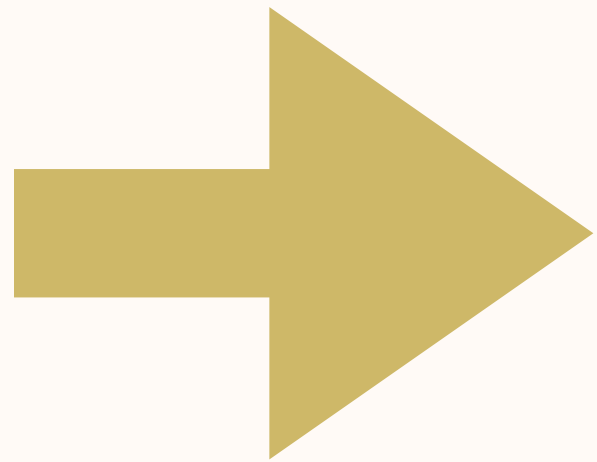
Un grand nombre de features

3 grandes classes de features quasi indépendantes

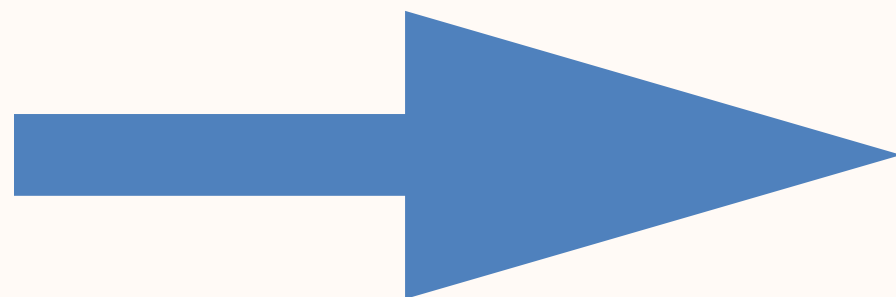
Une distribution inégale intra classe

Défis du dataset

Un grand nombre de features



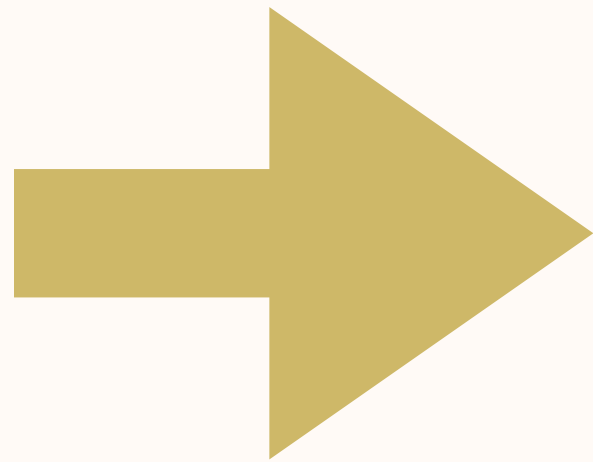
Espace de données complexe
Difficulté computationnelle
Risque de surapprentissage



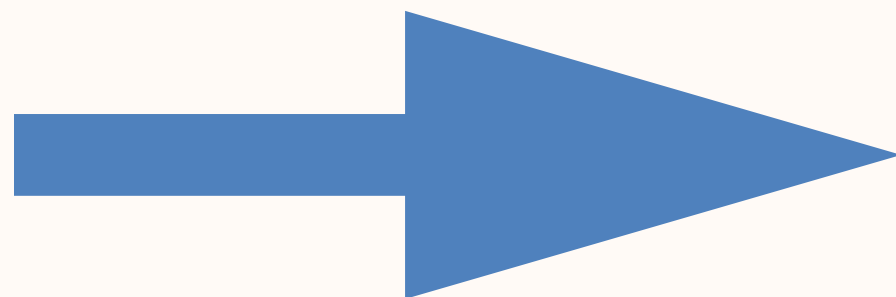
Réduire le nombre de features ?
Utiliser un modèle adapté aux espaces de données complexe ?

Défis du dataset

Un dataset divisé en 3



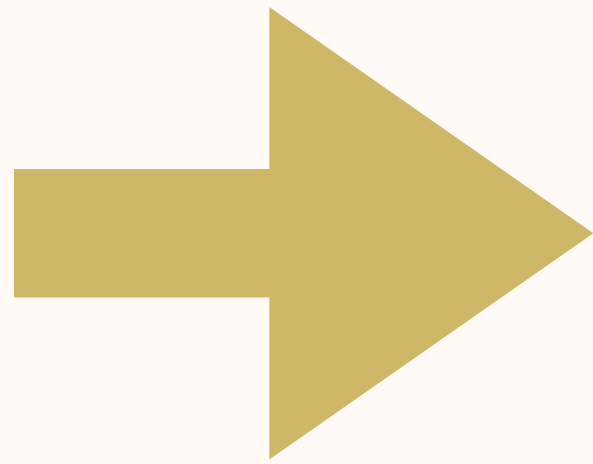
Difficulté pour un classificateur classique à classifier 3 jeux de données qui paraissent indépendants entre eux



Utiliser 3 modèles en parallèle ?
Enrichir les relations interclasses ?
Utiliser un modèle qui puisse traiter le dataset avec une attention multi-head ?

Défis du dataset

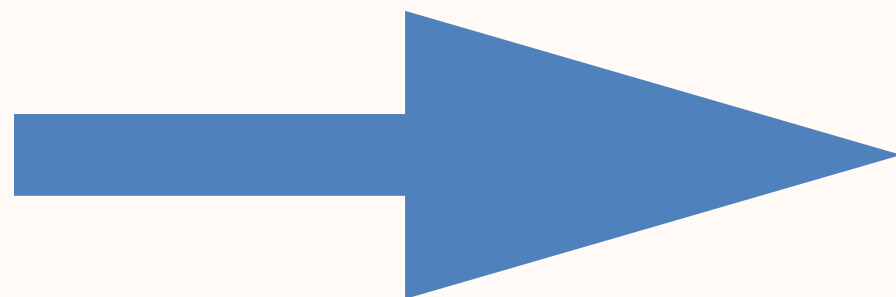
Distribution inégale intra classe



Les tags prédominants écraseront les tags rares pendant l'entraînement

Mauvaise performance pour les tags rares

Biais privilégiant les tags dominants



Effectuer une pondération ?

Enrichir le dataset avant classification ?

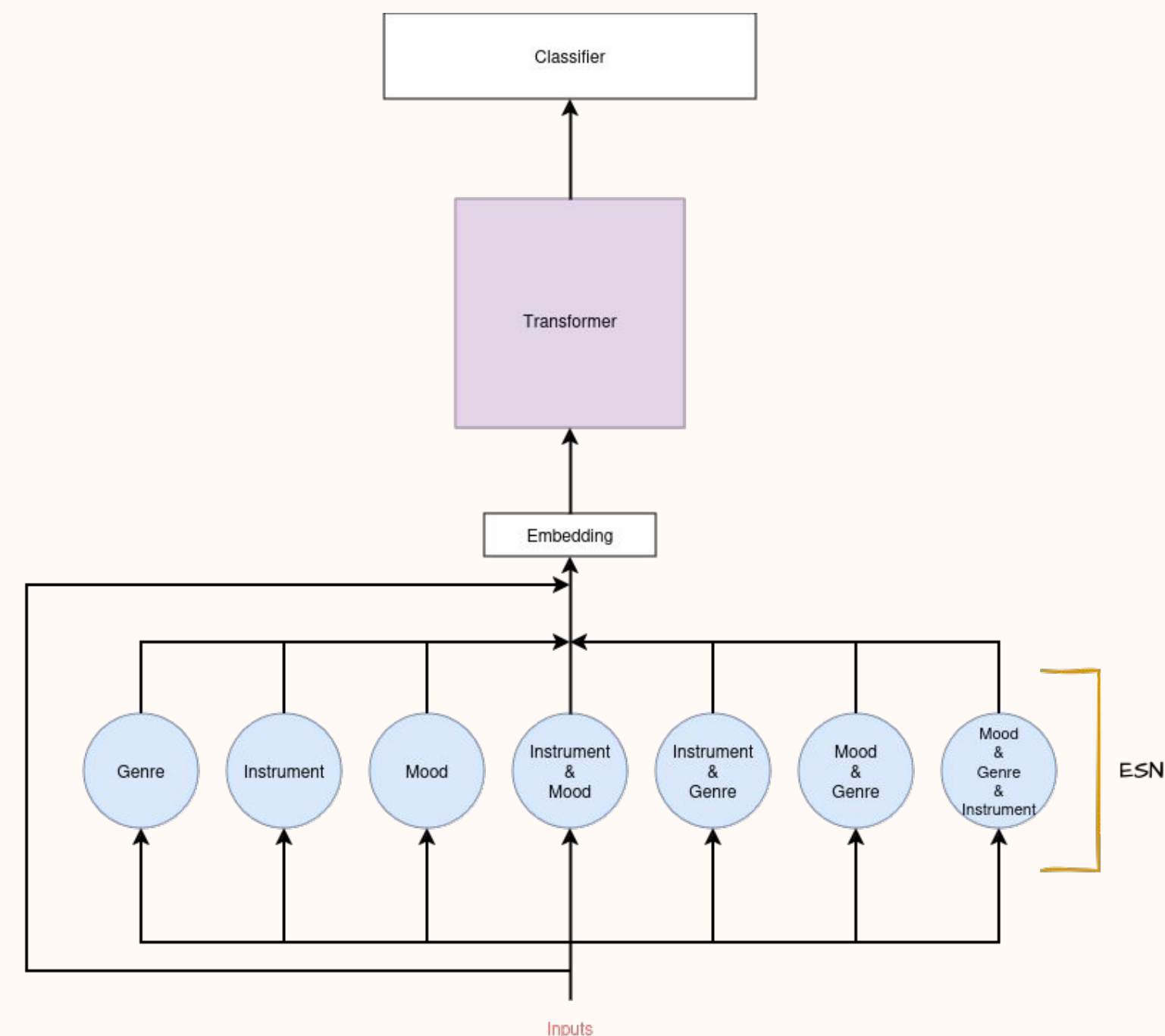
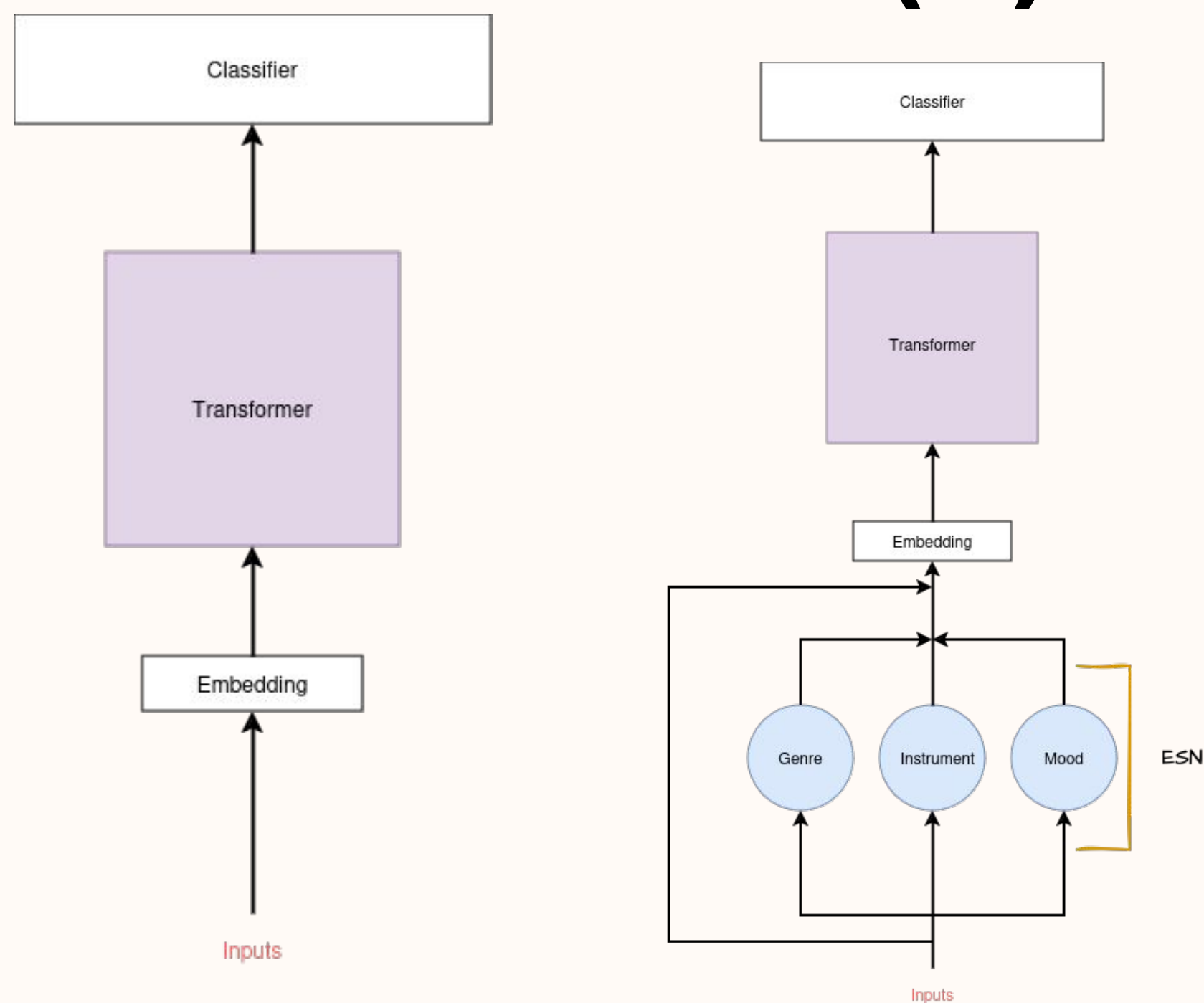
Utiliser un modèle avec une attention non uniforme ?

Plusieurs solutions envisagées

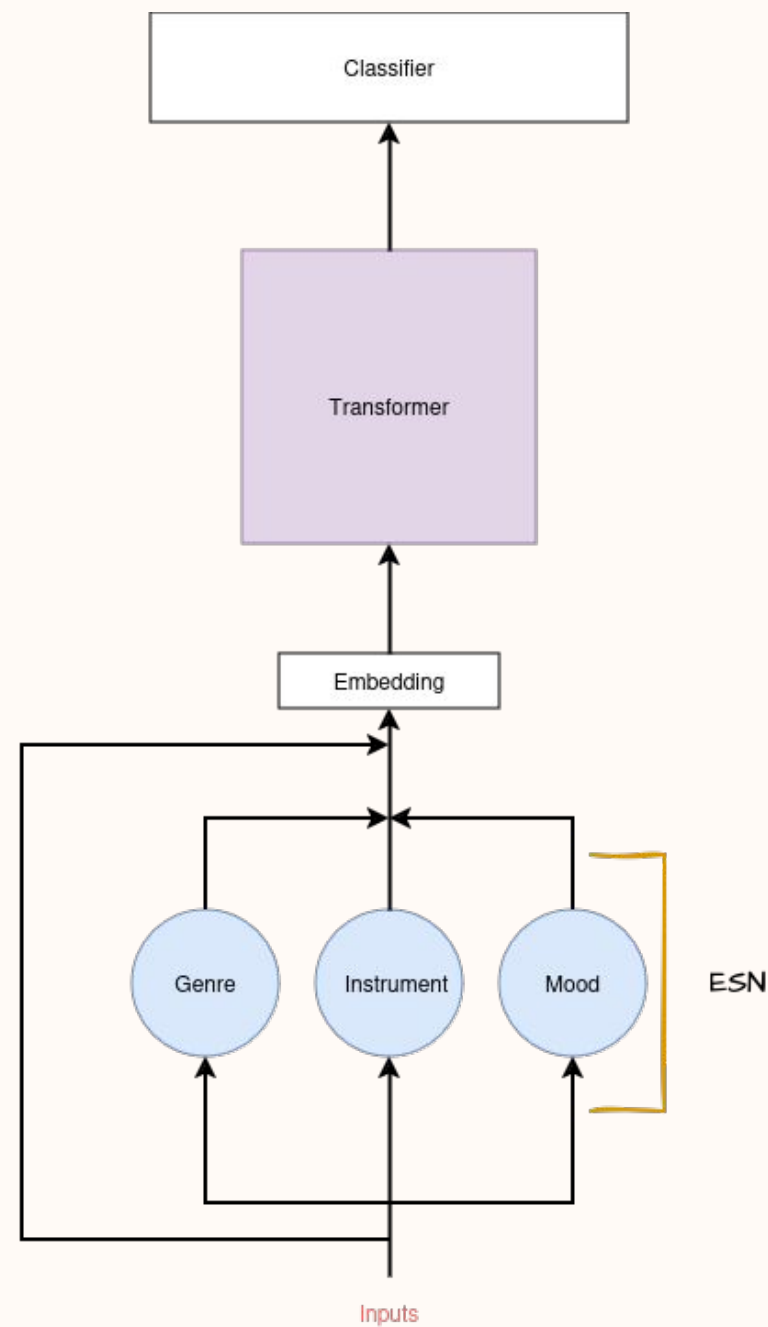


Random Forest	Distribution inégale 3 classes non corrélées	Espace de données peut-être trop complexe
Transformer	Gestion d'espace de données complexe 3 classes non corrélées	Tendance au sur-apprentissage sur données déséquilibrées
ESN	Enrichir les relations intra/inter classes	Pas d'autres avantages

Architecture(s)



Architecture retenue



Spécificités du modèle :

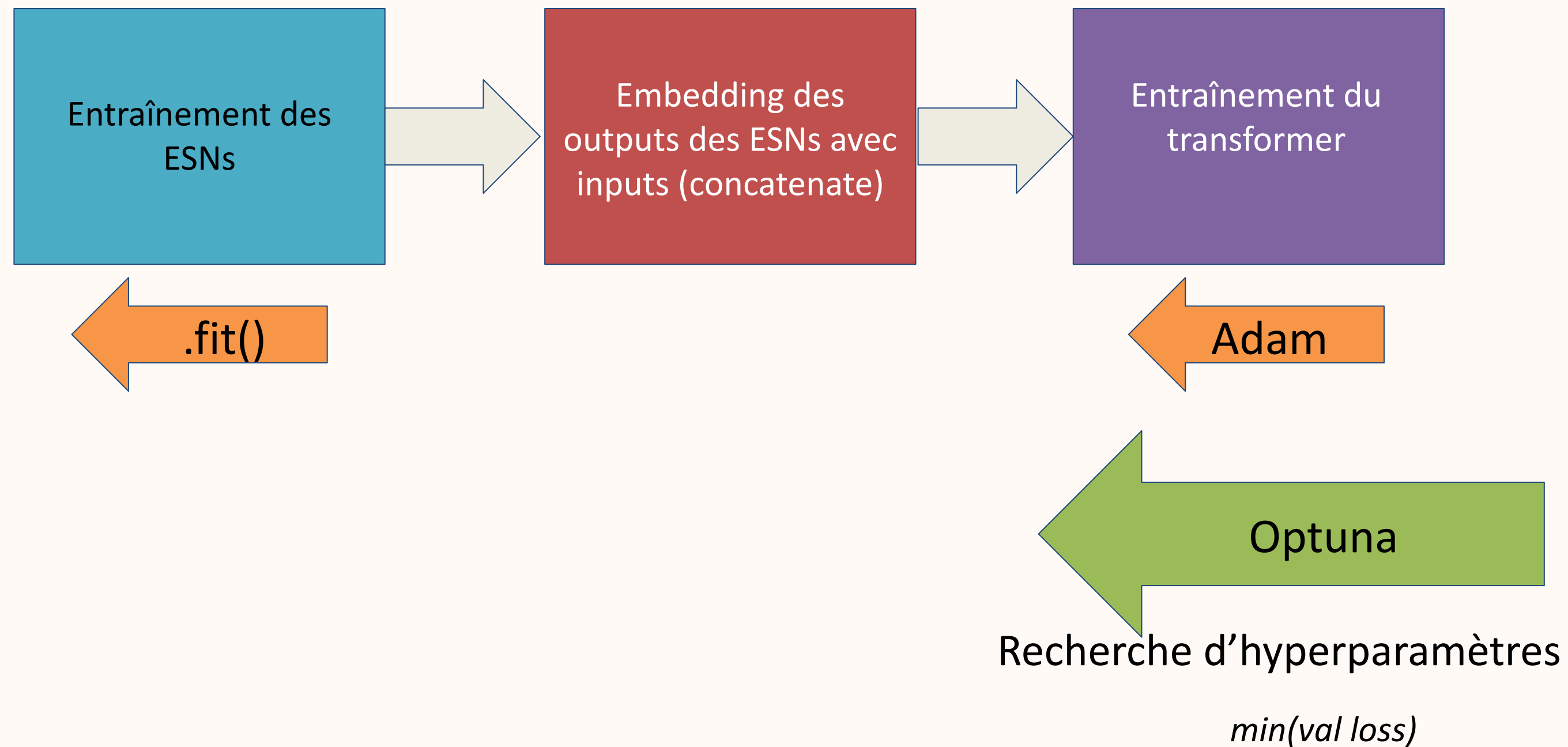
reservoirpy :

- Réservoirs de neurones paramétrés presque comme des Extreme Learning Machine (sans récurrence)

pytorch :

- Embedding qui concatène et compresse tous les inputs ensemble (nn.Linear)
- Transformer (nn.Transformer)
- Classifieur (nn.Linear)

Entraînement



```

run_model.ipynb M  train_transformer_final.ipynb M X
music_catalogs_classifier > train_transformer_final.ipynb > import torch

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ...
.venv (Python 3.12.3)

import torch
import torch.nn as nn
import torch.optim as optim
from reservoirpy.nodes import Reservoir, Ridge, ESN
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd

[1] ✓ 1.7s Python

# Configuration
DEVICE = torch.device("cuda" if torch.cuda.is_available() else "cpu")
BATCH_SIZE = 64
# EMBEDDING_SIZE = 256 # Taille des embeddings pour tous les inputs
# NUM_HEADS = 8
# NUM_LAYERS = 4
# DROPOUT = 0.1
# LEARNING_RATE = 1e-4
EMBEDDING_SIZE = 285
NUM_HEADS = 15
NUM_LAYERS = 2
DROPOUT = 0.2618574215322765
LEARNING_RATE = 0.00015775441092497324
EPOCHS = 20
PATIENCE = 3

[2] ✓ 0.0s Python

y_instruments = pd.read_csv("../data/train/output_instruments_tags_data.csv")
y_moods = pd.read_csv("../data/train/output_moods_tags_data.csv")

# On peut garder seulement une partie des données
X_genres = X_genres[: int(partOfData * len(X_genres))]
X_instruments = X_instruments[: int(partOfData * len(X_instruments))]
X_moods = X_moods[: int(partOfData * len(X_moods))]
y_genres = y_genres[: int(partOfData * len(y_genres))]
y_instruments = y_instruments[: int(partOfData * len(y_instruments))]
y_moods = y_moods[: int(partOfData * len(y_moods))]
X_genres_categories = X_genres_categories[
    : int(partOfData * len(X_genres_categories))
]
X_instruments_categories = X_instruments_categories[
    : int(partOfData * len(X_instruments_categories))
]

[3]

```

Optimisation des hyperparamètres

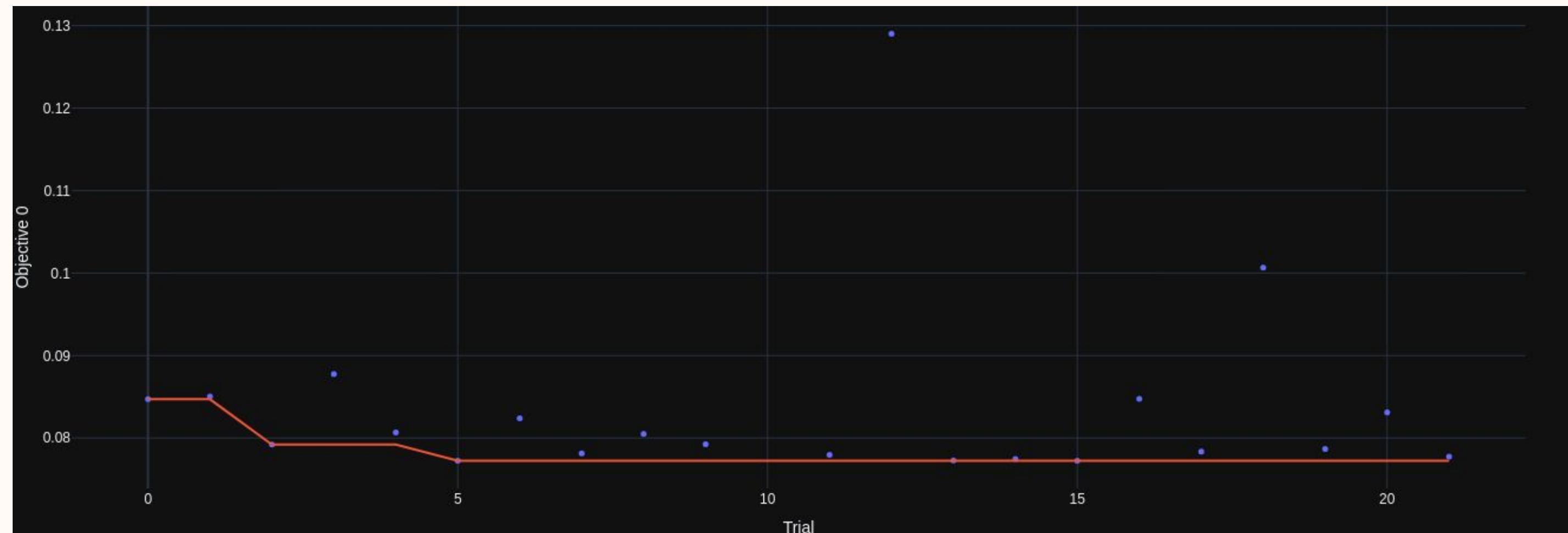
Best Trial (number=5)

0.07725657431556167

Params = [num_heads: 15, embedding_size: 285, num_layers: 2, dropout: 0.2618574215322765, learning_rate: 0.00015775441092497324]



OPTUNA



Résultats

Rang	Date	Participant(s)	Score public
1	4 avril 2021 21:11	gcanat	0,4785
2	21 novembre 2021 21:30	aho	0,4651
3	12 avril 2021 03:17	huy217	0,4650
4	15 décembre 2021 20:53	vincent.bour	0,4646
5	28 janvier 2021 22:50	cakedev	0,4585
6	12 décembre 2021 09:40	luguedon & BaptisteBenard	0,4575
7	16 décembre 2021 12:35	ppavia & enthomas	0,4568
8	17 décembre 2021 16:55	mkouhou & Maelle_a	0,4563
9	7 avril 2021 16:42	yfe	0,4562
10	8 mars 2021 18:24	Dupin_Sylvio	0,4562
11	12 novembre 2021 10:23	serrabil & slebdaoui	0,4562
12	-	benchmark	0,4561

notre score : ~ 0.495

Rapidité du modèle :
8h pour analyser le catalogue
complet de Spotify (100 millions
de titres)

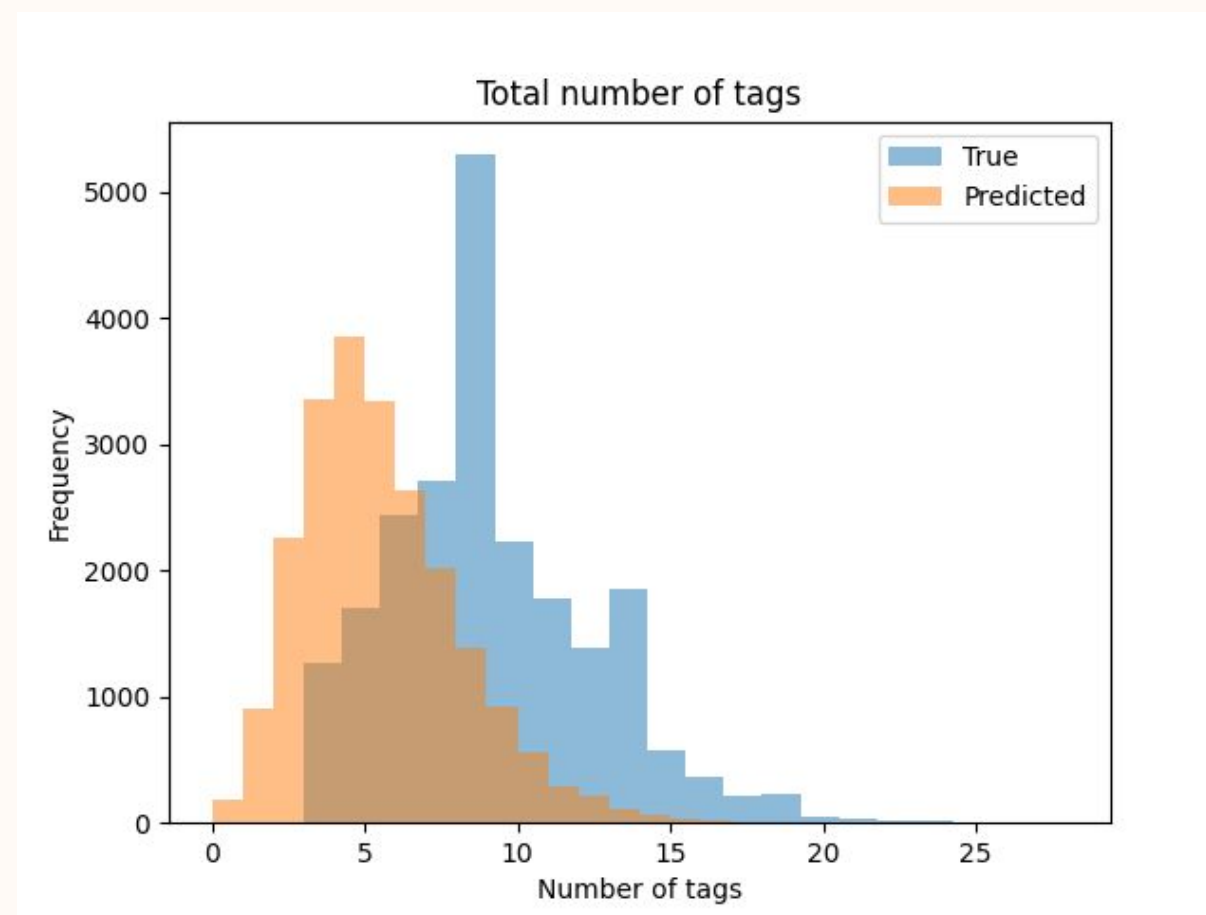
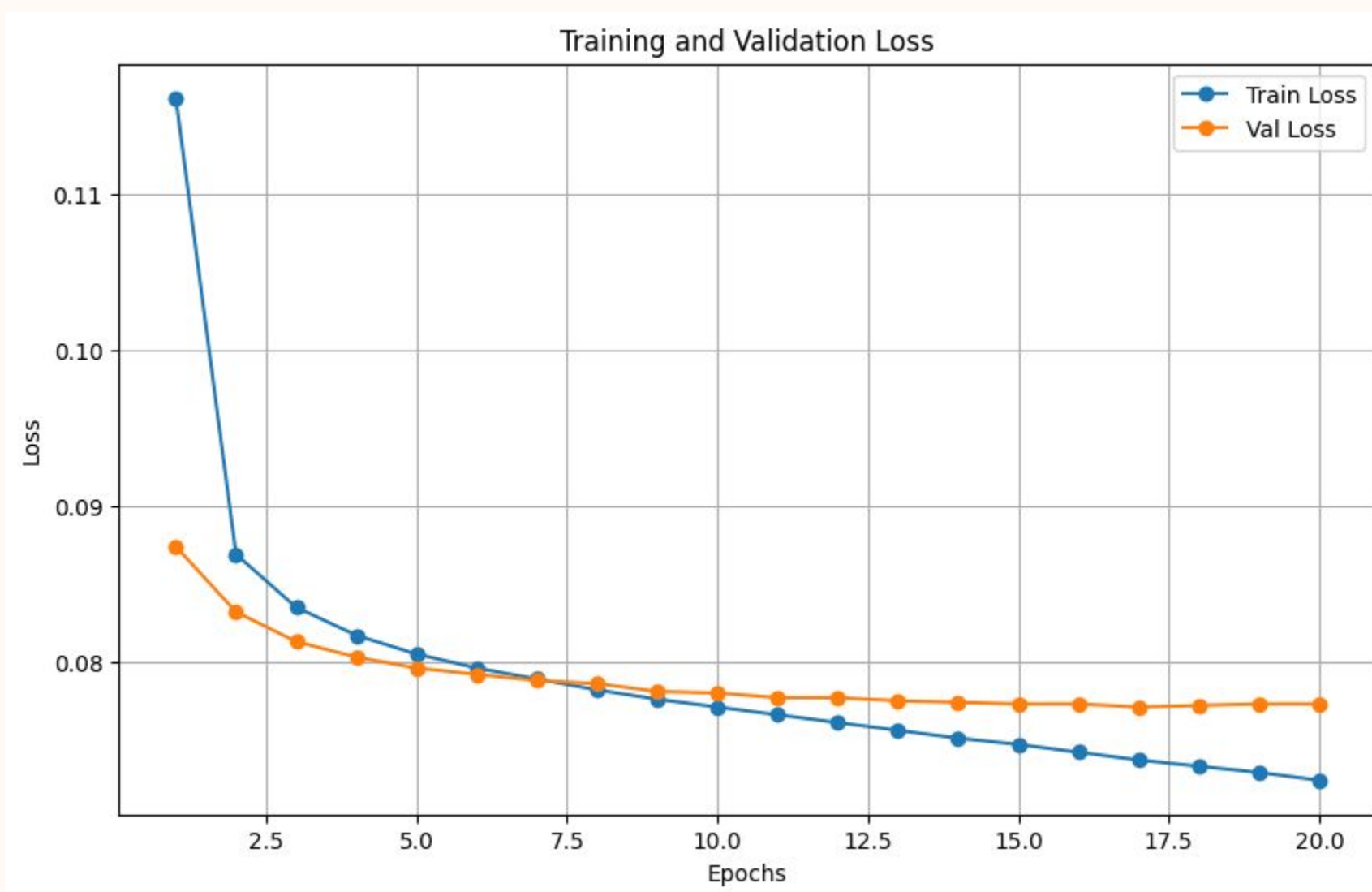
Résultats

Impact des ESNs

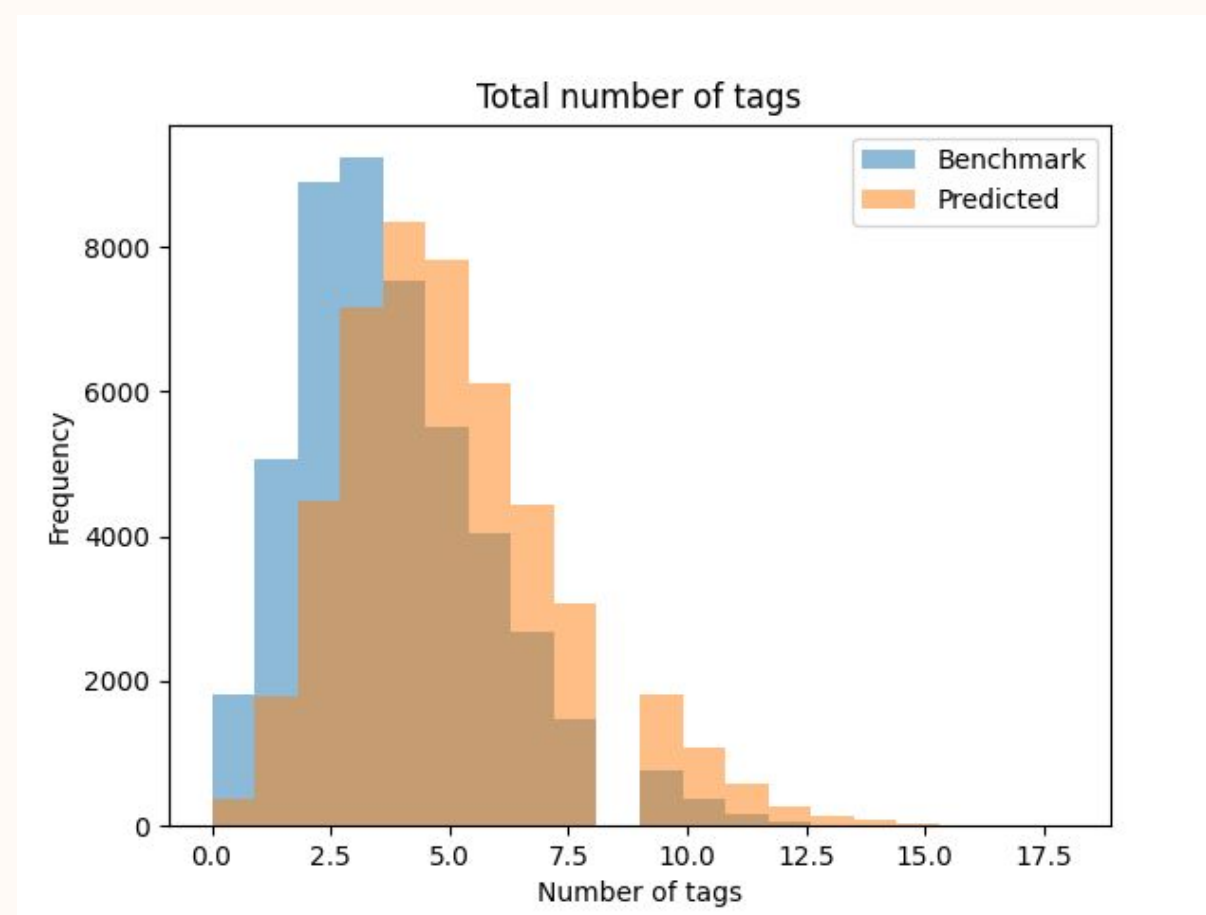
	Avec ESN	Sans ESN
Weighted avg F1 Score	0.495	0.484
Accuracy	0.97	0.97
Test Loss	0.0777	0.0784

Résultats

Train dataset:



Test dataset:



Résultats

Song 1:

Benchmark Tags: chamber-orchestra, clarinet, glockenspiel, pizzicati, relaxed, warm

Predicted Tags: chamber-orchestra, clarinet, glockenspiel, pizzicati, woodwinds-ensemble, relaxed, warm

Song 2:

Benchmark Tags: indie-rock, bass-guitar, drum-kit, electric-guitar, rock-band, driving, spectacular

Predicted Tags: indie-rock, acoustic-guitar, bass-guitar, drum-kit, electric-guitar, tambourine, confident, driving, spectacular

Song 3:

Benchmark Tags: contemporary-classical, confident, optimistic

Predicted Tags: confident, epic, inspirational, optimistic, uplifting

Song 4:

Benchmark Tags: contemporary-classical, symphony-orchestra, majestic

Predicted Tags: contemporary-classical, military-march, marching-band, symphony-orchestra, epic

Song 5:

Benchmark Tags:

Predicted Tags: world

Song 6:

Benchmark Tags: hard-rock, heavy-metal, bass-guitar, drum-kit, electric-guitar, male-vocals, rock-band, vocal, confident, driving

Predicted Tags: electric-guitar, male-vocals, vocal, confident, driving

Song 7:

Benchmark Tags: sinister, suspenseful

Predicted Tags: drones, electronica, serious, sinister, suspenseful

Song 8:

Benchmark Tags: bass-guitar, drum-kit, electric-guitar

Predicted Tags: bass-guitar, drum-kit, electric-guitar, synth-pad, synthesizer, spectacular

Répartition

Axel

Préparation et analyse des
données
Code réservoirs

Elouarn

Code Transformer
Code global
Entrainement

Réflexion sur les modèles

Retour sur le projet

Une architecture **farfelue** pour tester et explorer le plus possible

Impact **minime** des ESN sur la performance du modèle

Satisfaits du projet même si on a pas pu avoir le jeu de données **y_test** privées du challenge

Horizon : modèle qui prend en entrée le fichier audio et retourne les probabilités de tags

Merci pour votre attention