

Classifieur de sons

IN104 2022

Contact et déroulement du projet

Contact : Gwendal.PRISER@ensta-paris.fr ou ENSTA : **R 2.20**

Repo GitHub contenant le projet : <https://github.com/gpU2IS/IN104>

Outils nécessaires :

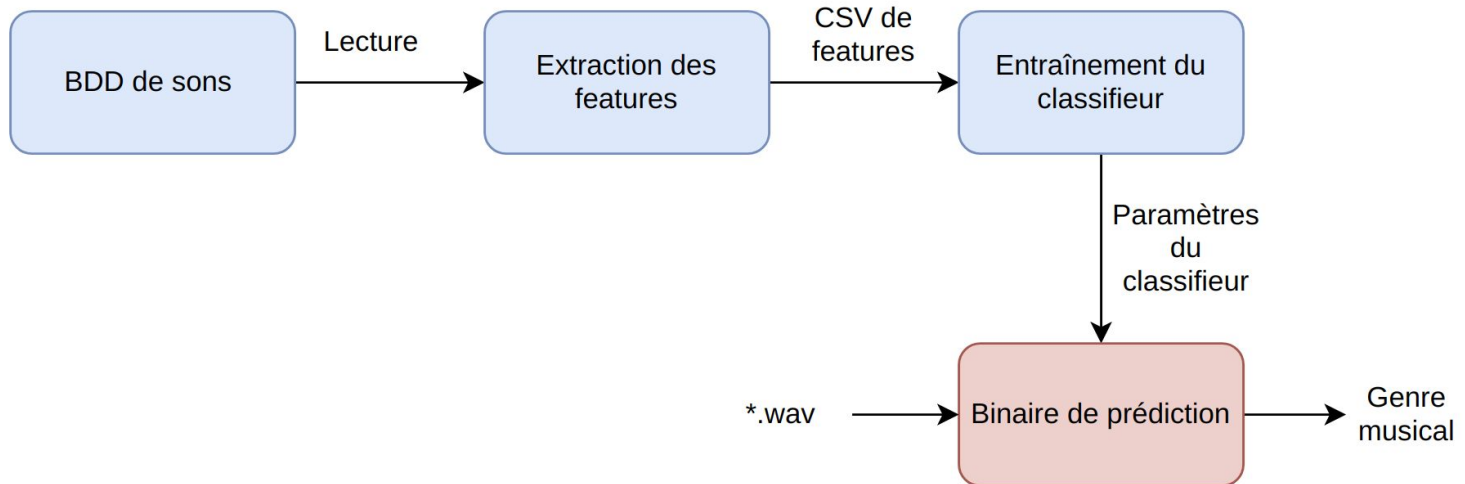
- **Linux** (Ubuntu, ...)
- Compilateur **GCC** (compilation du programme C)
- **Python 3** (entraînement d'un classifieur, outils de visualisation)
- Connexion internet

mardi 22/03	14:45 - 15:45		Cours magistral
mardi	16:00 - 18:00		TD info
mardi 29/03	14:45 - 18:00		TD info
mardi	16:00 - 18:00		TD info
mardi 05/04	14:45 - 16:45		cours magistral
mardi 12/04	14:45-16:45		TD info
mardi 19/04	14:45-16:45		TD info
mardi 3/05	14:45-16:45		TD info
mardi 10/05	14:45-16:45		TD info
mardi 24/05	14:45 - 18:00		Soutenance 2

Projet

Objectif : Création d'un programme C classifiant des sons en genre musicaux.

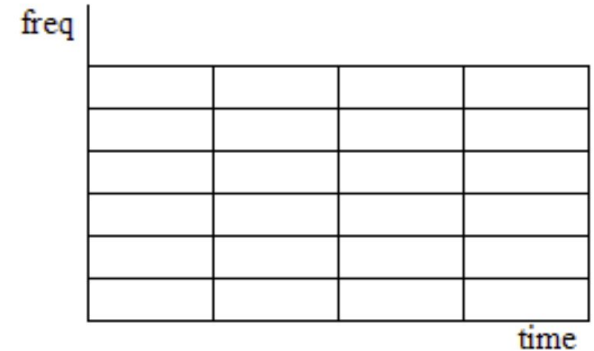
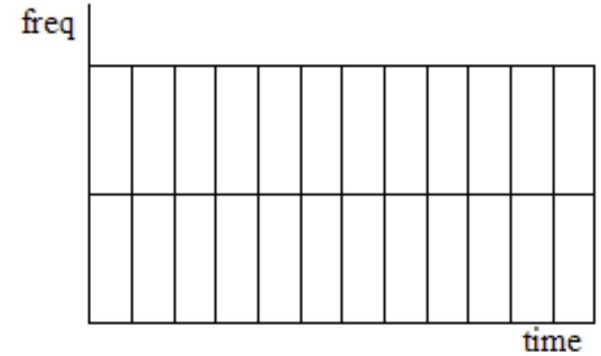
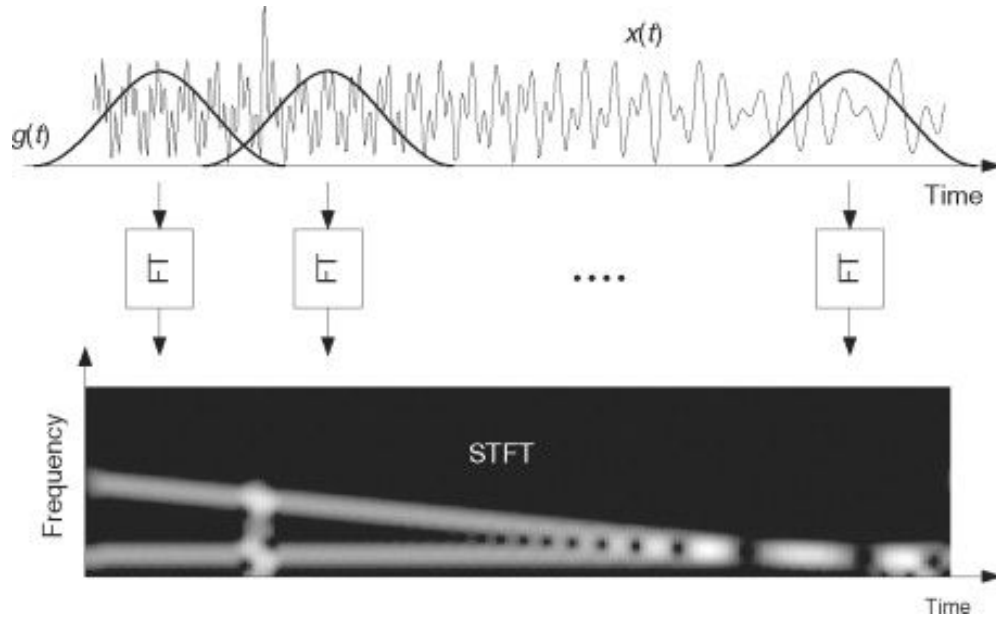
Etapes du projet :



(<http://marsyas.info/downloads/datasets.html>)

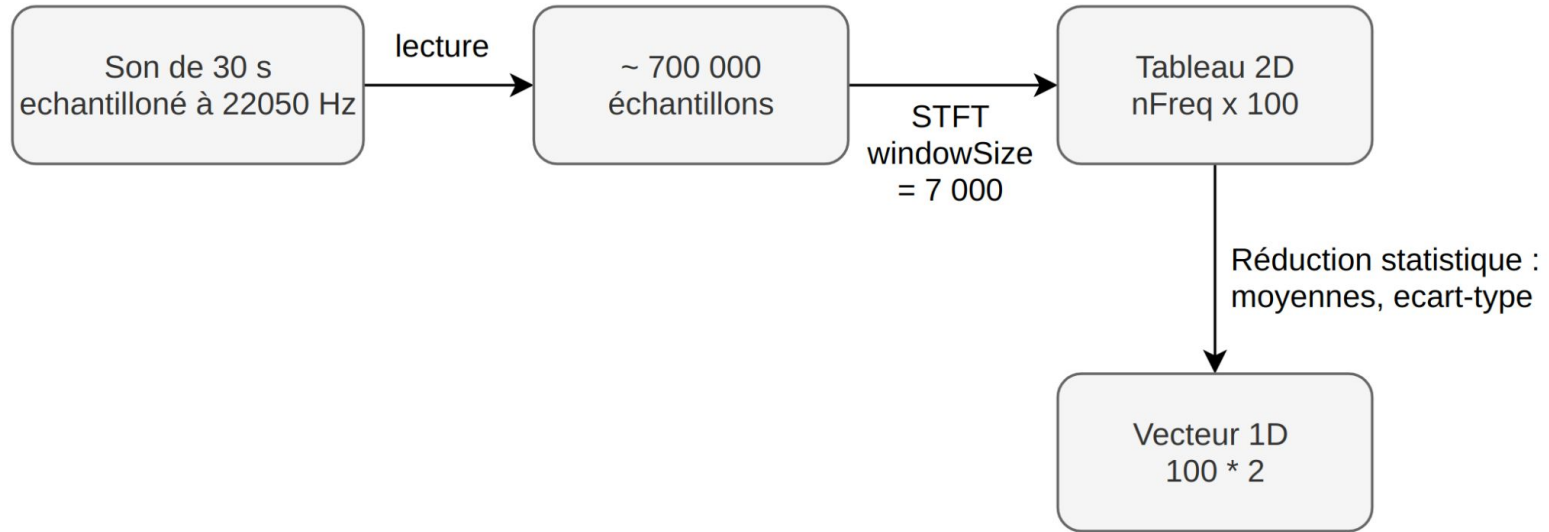
Extraction des descripteurs (1)

Short - Time - Fourier - Transform



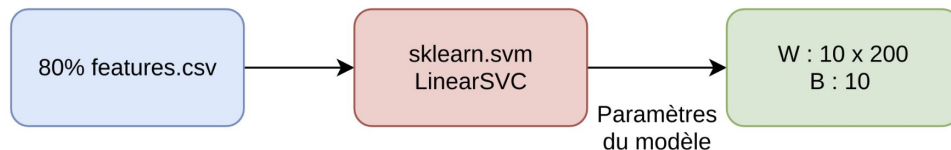
(<https://www.sciencedirect.com/topics/engineering/short-time-fourier-transform>)

Extraction des descripteurs (2)



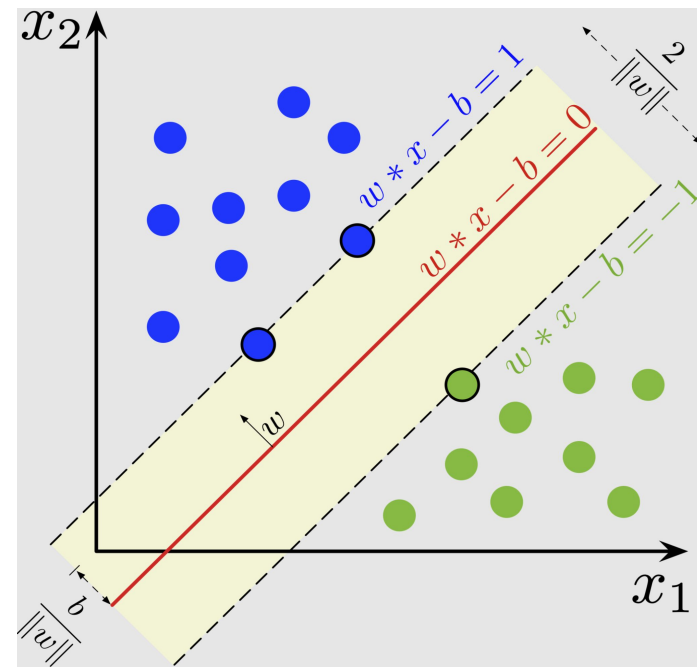
Apprentissage supervisé et classifieur

$$\min_{\theta} \frac{1}{N} \sum_{i=0}^N (f_{\theta}(x_i) - y_i)^2$$

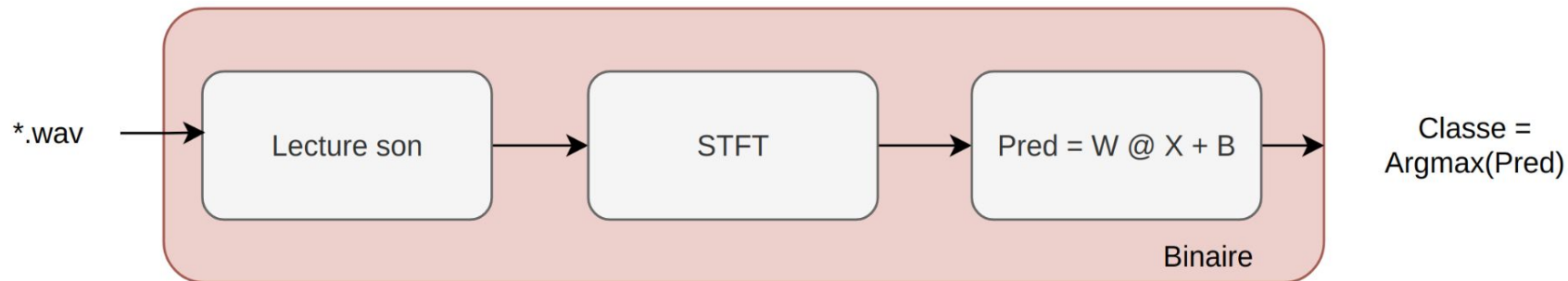


features.csv :

```
0; mu1; signal; ... ; mu200; sigma200;  
.  
.  
.  
9; mu1; signal; ... ; mu200; sigma200;  
  
(100 x 10 lignes, 200 + 1 colonnes)
```



Prédiction



```
~/IN104 ./classify song.wav
This song is classical music%
```

Présentation des résultats

- Quelles sont les données utilisées ?
- Matrice de confusion
- Temps d'entraînement
- Temps d'inférence

True Class	airplane	923	4	21	8	4	1	5	5	23	6
	automobile	5	972	2					1	5	15
	bird	26	2	892	30	13	8	17	5	4	3
	cat	12	4	32	826	24	48	30	12	5	7
	deer	5	1	28	24	898	13	14	14	2	1
	dog	7	2	28	111	18	801	13	17		3
	frog	5		16	27	3	4	943	1	1	
	horse	9	1	14	13	22	17	3	915	2	4
	ship	37	10	4	4		1	2	1	931	10
	truck	20	39	3	3			2	1	9	923
		airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
		Predicted Class									

Architecture du projet

```
/IN104_Groupe
```

```
src/
```

```
main.c
```

```
include/ (modules du projet)
```

```
*.h
```

```
*.c
```

```
docs/
```

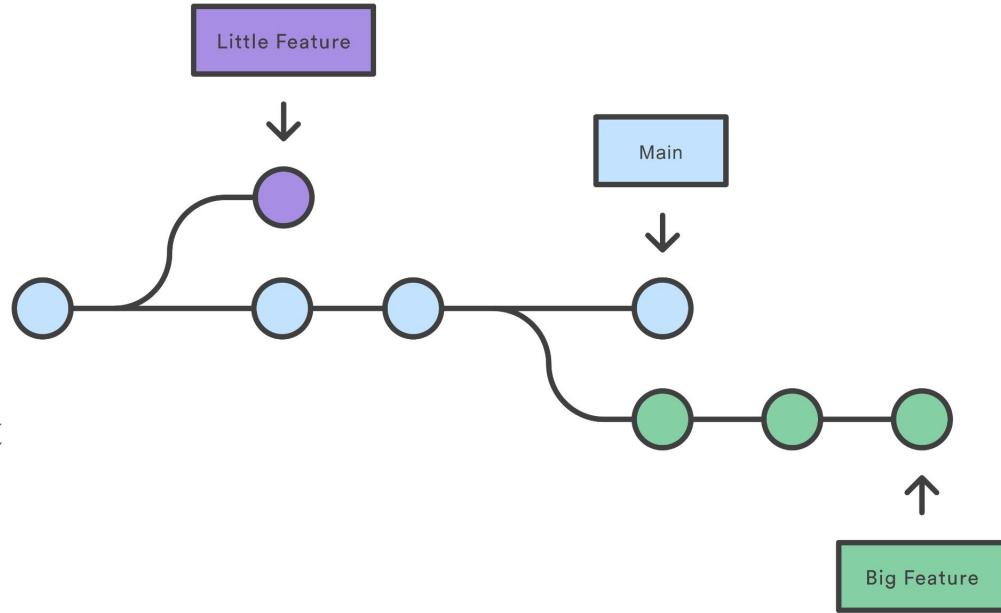
```
examples/
```

```
Makefile
```

```
README.md (Présentation du projet, instruction pour la compilation, comment  
exécuter le programme ?, résultats obtenus)
```

Utilisation de Git

```
git init  
git add <files>  
git commit -m "<message descriptif>"  
git remote add origin  
git@github.com:<user>/<reponame>.git  
git push origin master.
```



Conseils

- Compiler son code et l'exécuter régulièrement.
- Valider les différents modules sur des exemples simples.
- Effectuer des commits réguliers.
- En groupe, effectuer une architecture logicielle en amont avec entrées / sorties des modules : respecter la convention de nommage. E.g. : `maVar`, `MaStruct`, `mon_module`, `CONSTANTE`.
- Utiliser un IDE (VSCode ...) pour pouvoir débbugger avec GDB.
- Effectuer ses recherches internet en anglais car plus de réponses sont disponibles.

Suivi et évaluation du projet

Tableau d'avancement :

https://docs.google.com/spreadsheets/d/1iR_tJWAhS-ZisnpE-PrqFb7A_0W4x8w9xwW4t1x4P6A/edit?usp=sharing

Soutenance

Date : Mardi 24 Mai de 14h45 à 18h

- Présentation des solutions techniques, des résultats ou pistes de recherche.
- Démonstration en direct : compilation du projet et inférence sur un exemple.

Durée maximale : 10 minutes