**Lab. 7. Gestionarea datelor audio folosind Python**

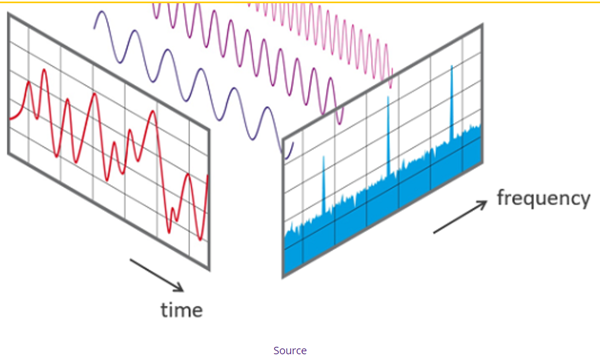
Sunetul este reprezentat sub forma unui semnal audio care are parametri precum frecvență, lățime de bandă, decibel etc.

Gestionarea fișierelor audio în Python: mp3, wav, m4a, flac.

Sunete digitale măsurate în frecvență (kHz) .

1 kHz = 1000 de informații pe secundă.

Un semnal audio tipic poate fi exprimat ca o funcție de amplitudine și timp.



Există dispozitive construite care vă ajută să surprindeți aceste sunete și să le reprezentați într-un format care poate fi citit de computer. Exemple de astfel de formate sunt format WAV (Waveform Audio File), format mp3 (MPEG-1 Audio Layer 3), Format WMA (Windows Media Audio).

Un proces tipic de procesare audio implică extragerea caracteristicilor acustice relevante pentru o sarcină, urmată de scheme de luare a deciziilor care implică detectarea, clasificarea și fuziunea cunoștințelor.

Există câteva biblioteci python utile care facilitează această sarcină.

Python are câteva biblioteci excelente pentru procesarea audio, cum ar fi Librosa și PyAudio.

Există, de asemenea, module încorporate pentru unele funcționalități audio de bază.

Vom folosi în principal două biblioteci pentru achiziționarea și redarea audio:

**1. Librosa**

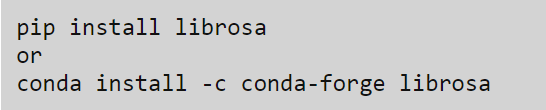
Este un modul Python pentru a analiza semnalele audio în general, dar orientat mai mult spre muzică. Acesta include toate elementele pentru a construi un sistem MIR (Music Information Retrieval).

A fost foarte bine documentat împreună cu o mulțime de exemple și tutoriale.

Python are câteva biblioteci excelente pentru procesarea audio, cum ar fi Librosa și PyAudio.

Există, de asemenea, module încorporate pentru unele funcționalități audio de bază.

Instalare:



**2. IPython.display.Audio**

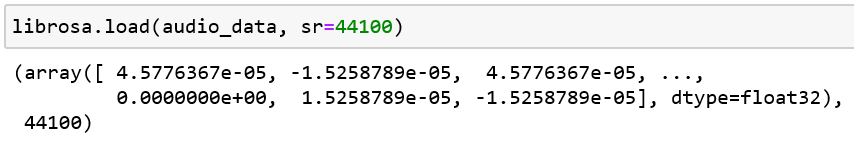
IPython.display.Audio vă permite să redați sunetul direct într-un notebook Jupyter.

Vocaroo | Înregistrator de voce online. Vocaroo este o modalitate rapidă și ușoară de a partaja mesaje vocale pe interwebs.

Încărcarea unui fișier audio:



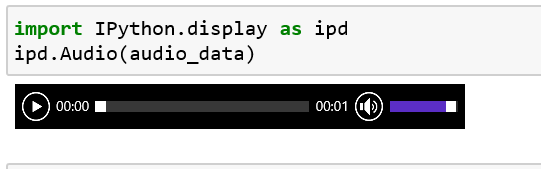
Aceasta returnează o serie de timp audio ca o matrice numpy cu o rată de eșantionare implicită (sr) de 22 kHz mono. Putem schimba acest comportament prin eșantionarea la 44,1 kHz.



Rata de eșantionare este numărul de eșantioane de sunet transportate pe secundă, măsurate în Hz sau kHz.

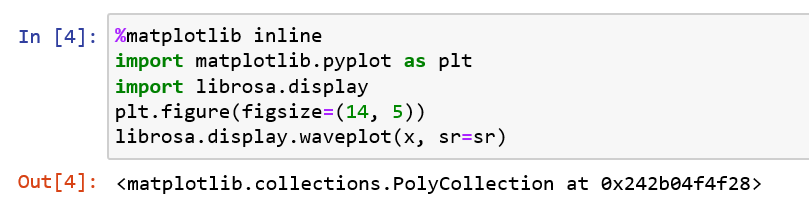
Redarea audio:

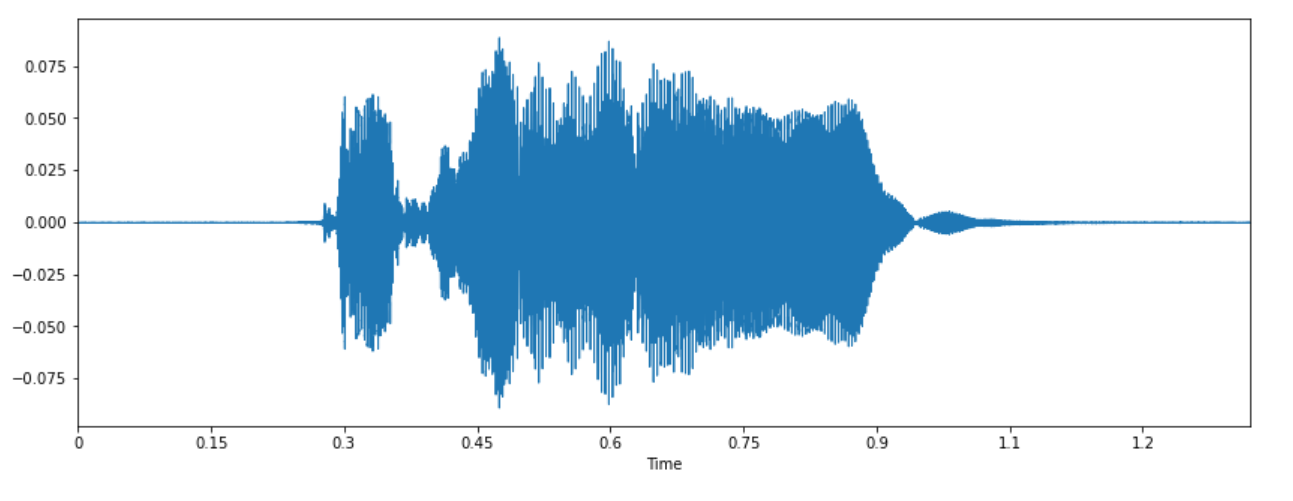
Folosind IPython.display.Audio puteți reda sunetul în notebook-ul Jupyter.



**Vizualizarea audio:**

Putem vizualiza matricea audio folosind librosa.display.waveplot:



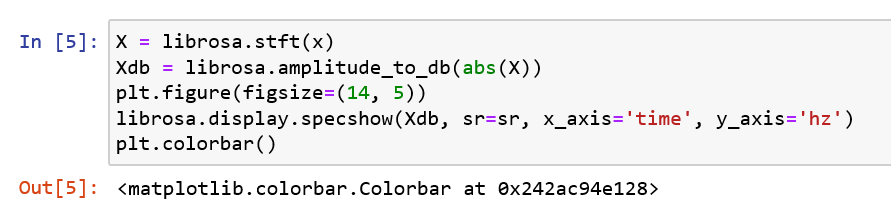


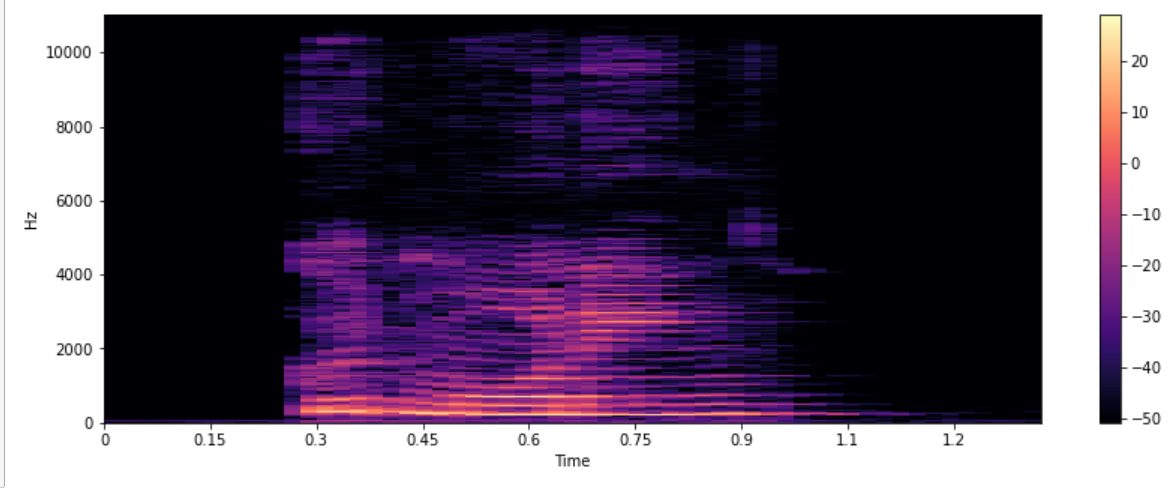
**Spectrogramă**

O spectrogramă este o modalitate vizuală de a reprezenta puterea semnalului sau „intensitatea” unui semnal în timp la diferite frecvențe prezente într-o anumită formă de undă. Nu numai că se poate vedea dacă există mai multă sau mai puțină energie la, de exemplu, 2 Hz față de 10 Hz, dar se poate vedea și modul în care nivelurile de energie variază în timp.

O spectrogramă este de obicei descrisă ca o hartă de căldură, adică ca o imagine cu intensitatea arătată prin variația culorii sau a luminozității.

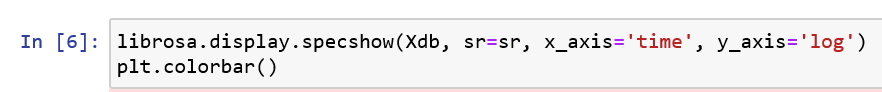
Putem afișa o spectrogramă folosind librosa.display.specshow.

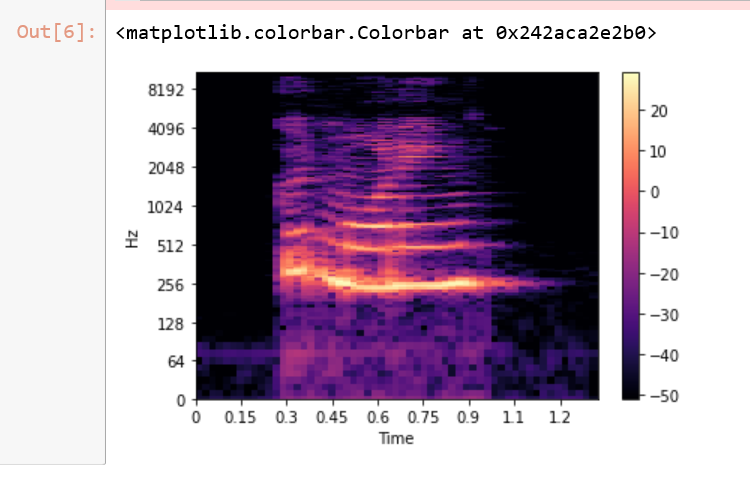




.stft () convertește datele în transformată Fourier pe termen scurt. STFT convertește semnale astfel încât să putem cunoaște amplitudinea frecvenței date la un moment dat. Folosind STFT putem determina amplitudinea diferitelor frecvențe redate la un moment dat al unui semnal audio. .specshow este folosit pentru a afișa o spectrogramă.

Axa verticală prezintă frecvențe (de la 0 la 10 kHz), iar axa orizontală arată ora clipului. Deoarece vedem că toate acțiunile au loc în partea de jos a spectrului, putem converti axa de frecvență într-una logaritmică.



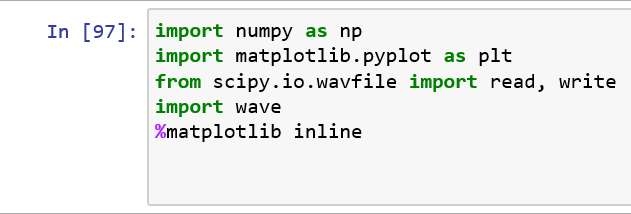


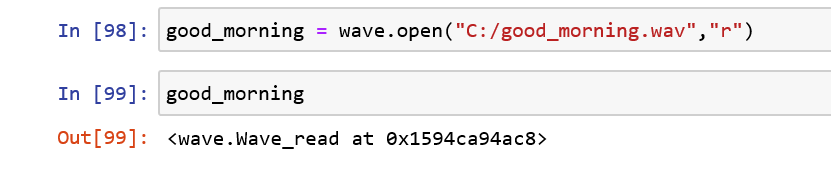
Exemple de frecvență

Melodiile în flux au o frecvență de 32 kHz

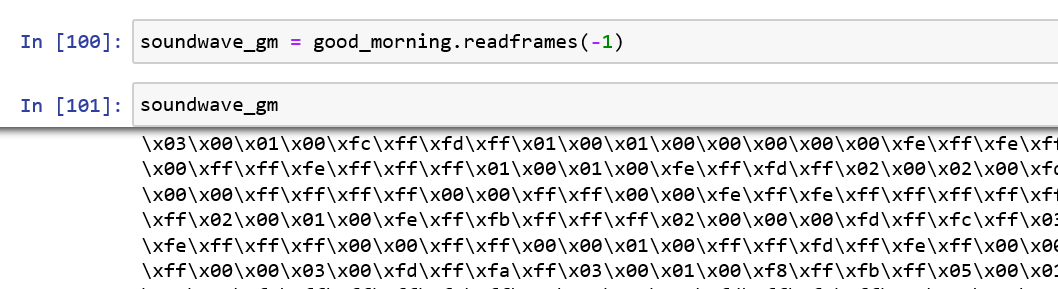
Cărțile audio și limbajul vorbit sunt cuprinse între 8 și 16 kHz

Nu putem vedea audio, așa că trebuie să le transformăm, mai întâi.





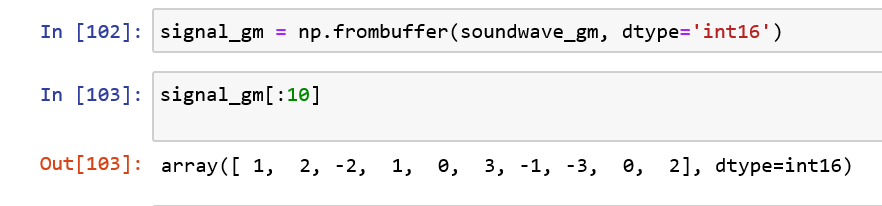
Convertirea obiectului wave in bytes și vizualizarea lui.



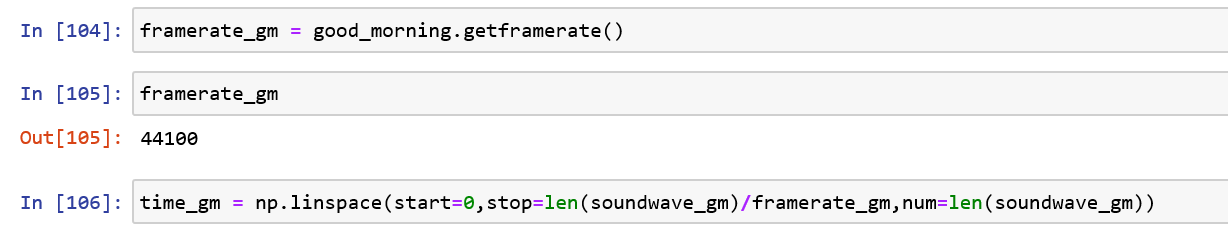
Trebuie să convertiți sunetul într-o formă utilizabilă.

Eșantion mic de audio = cantitate mare de informații.

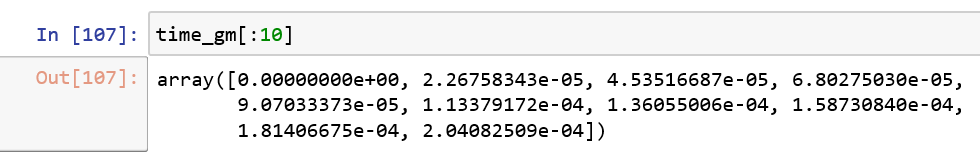
Conversia octeților în numere întregi. Nu se pot utiliza octeți. Convertiți octeți în numere întregi folosind numpy.



Obțineți marcajele temporale ale undei sonore



Vizualizarea primelor 10 marcaje temporale



Temă.

Implemetați codul si pentru fișiere: funky.wav, good-afternoon.wav, shuff89.wav, mrsandman.wav.