Proiect Python

„Analiza unui fișier audio și determinarea intervalelor de timp unde discuția devine încinsă”

Autor: Cătălin Banciu

# Introducere

Prezentul raport explică raționamentul autorului în rezolvarea sarcinii date: analiza unui fișier audio și determinarea intervalelor de timp unde discuția devine încinsă.

Salut, sunt Cătălin Banciu și am absolvit track-ul de Web Development Intermediar în 2024, fiind de asemenea Ambasador Generația Tech. Recent, am urmărit cursuri de Python pe YouTube (oferite în special de *BroCode*), completate de informațiile structurate de pe <https://www.w3schools.com/>

Stăpânesc limbajul de programare Python la un nivel junior. Pentru prezentul proiect, am folosit librăria de Python *librosa*, care mi s-a părut a fi cea mai potrivită, având în vedere sarcina dată.

Identificarea intervalelor de timp unde discuția devine încinsă a fost realizată în trei moduri diferite:

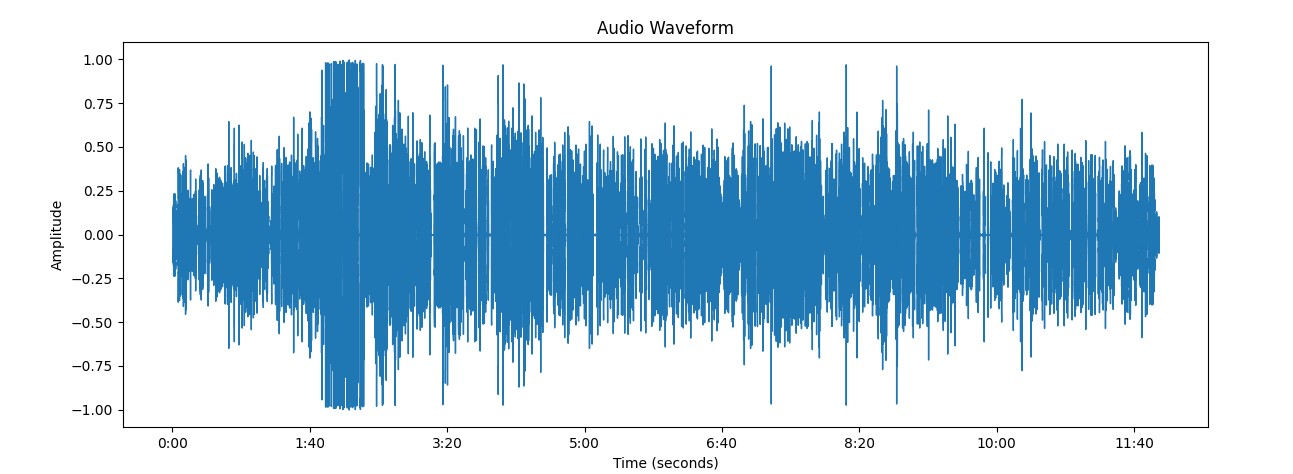
1. Folosind amplitudinea semnalului pe diferite porțiuni din înregistrare
2. Folosind o „învelitoare” (*envelope*) aplicată pe semnal
3. Folosind spectrul de frecvențe ale fișierului audio, determinat cu ajutorul transformatei Fourier

Fiecare abordare este scrisă într-un fișier .py, iar intervalele de timp sunt exportate în format .csv (într-un fișier corespunzător, distinct pentru fiecare abordare)

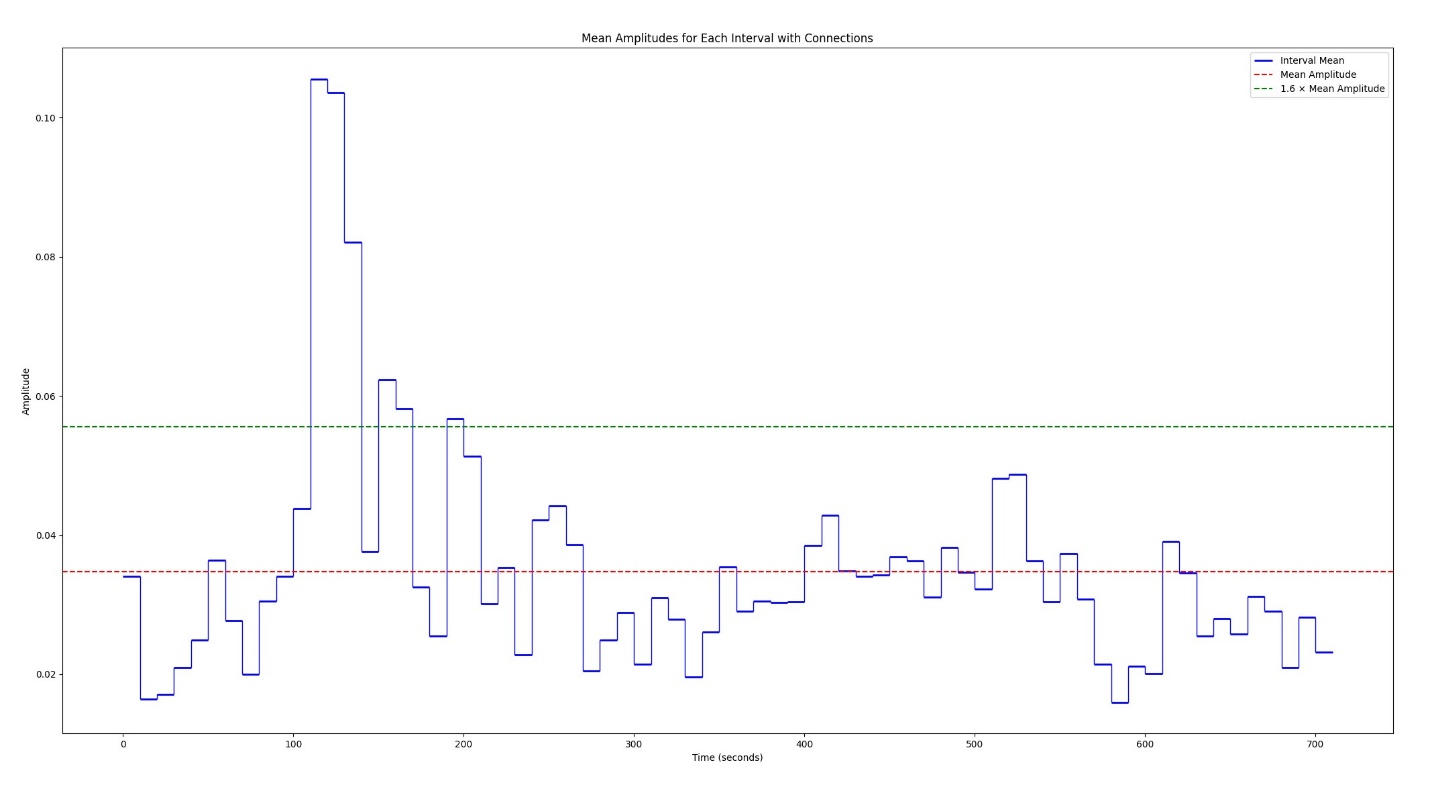
# Abordarea nr. 1

Mai întâi, video-ul cu Gigi Becali a fost descărcat de pe YouTube în format audio (.mp3). Audio-ul a fost apoi citit folosind *librosa.load*.

Graficul semnalului audio este redat în figura de pe pagina următoare:



Am calculat media globală a amplitudinii semnalului, pe întreaga durată a înregistrării. Apoi, am împărțit semnalul în secvențe de 10 secunde (durata aceasta poate fi ajustată). Pe fiecare diviziune, am calculat amplitudinea medie a semnalului. Dacă amplitudinea este mai mare de *k* ori decât media globală, intervalul este considerat „gălăgios”. În graficul de mai jos, am luat *k* = 1,6. Astfel, dacă media intervalului este de cel puțin 1,6 ori mai mare decât media globală, intervalul este marcat ca fiind probabil o „încingere” a discuției.



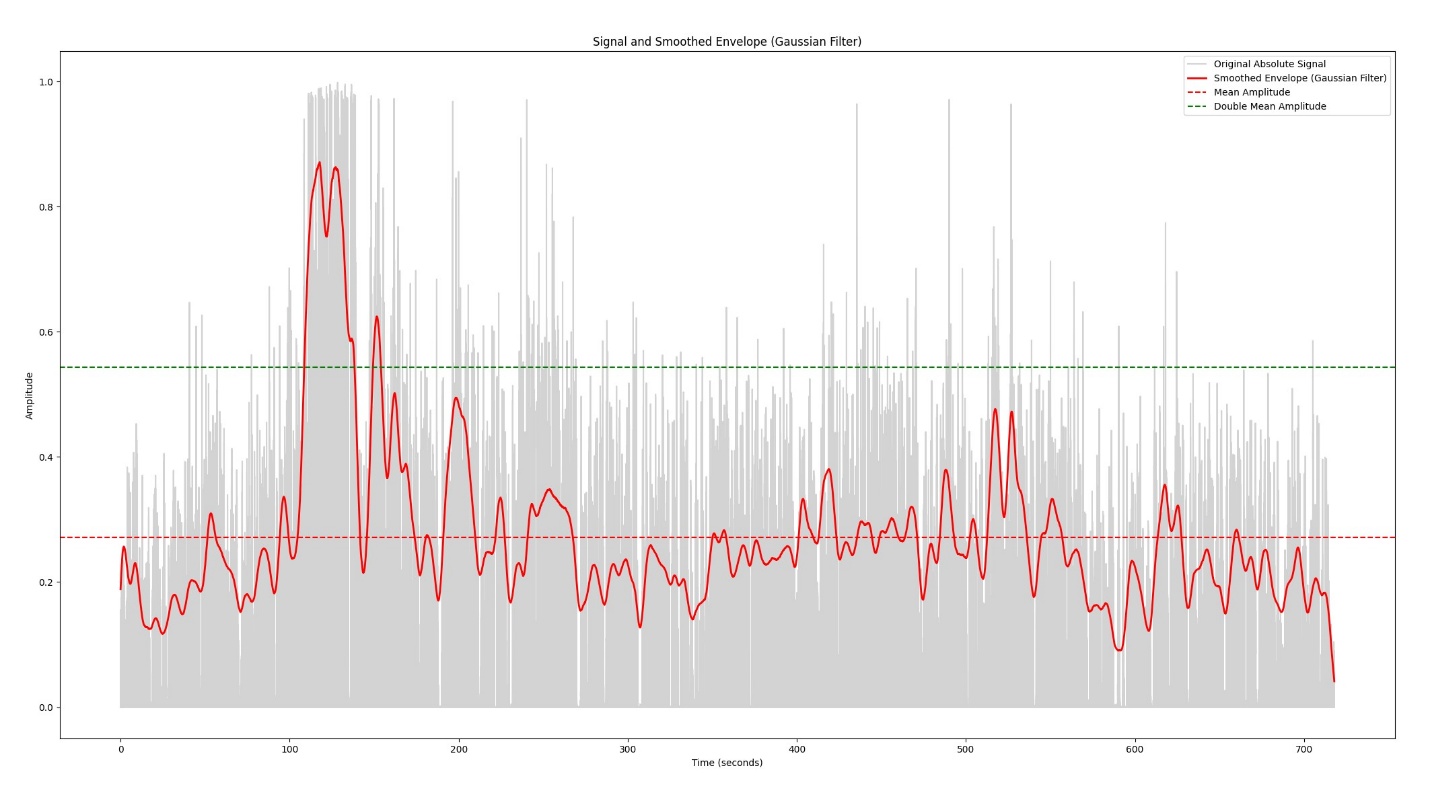
Intervalele rezultate sunt exportate într-un fișier .csv (time\_ranges.csv)

Această metodă este relativ rapidă și versatilă, oferind posibilități de ajustare prin prezența unor parametri (lungimea intervalului de referință, pragul *k* pentru amplitudine înaltă)

# Abordarea nr. 2

Mai întâi, video-ul cu Gigi Becali a fost descărcat de pe YouTube în format audio (.mp3). Audio-ul a fost apoi citit folosind *librosa.load*.

Cu ajutorul transformatei lui Hilbert și a unui filtru gaussian, am calculat învelitoarea semnalului (*signal envelope*). Aceasta este reprezentată în graficul de mai jos:



Învelitoarea arată trendurile amplitudinii, fără a oscila atât de rapid și haotic precum semnalul propriu-zis (*smoothed envelope*). Atunci când învelitoarea depășește un prag, notăm intervalul ca fiind „încins”. Pe graficul de mai sus, pragul este dublul mediei învelitoarei.

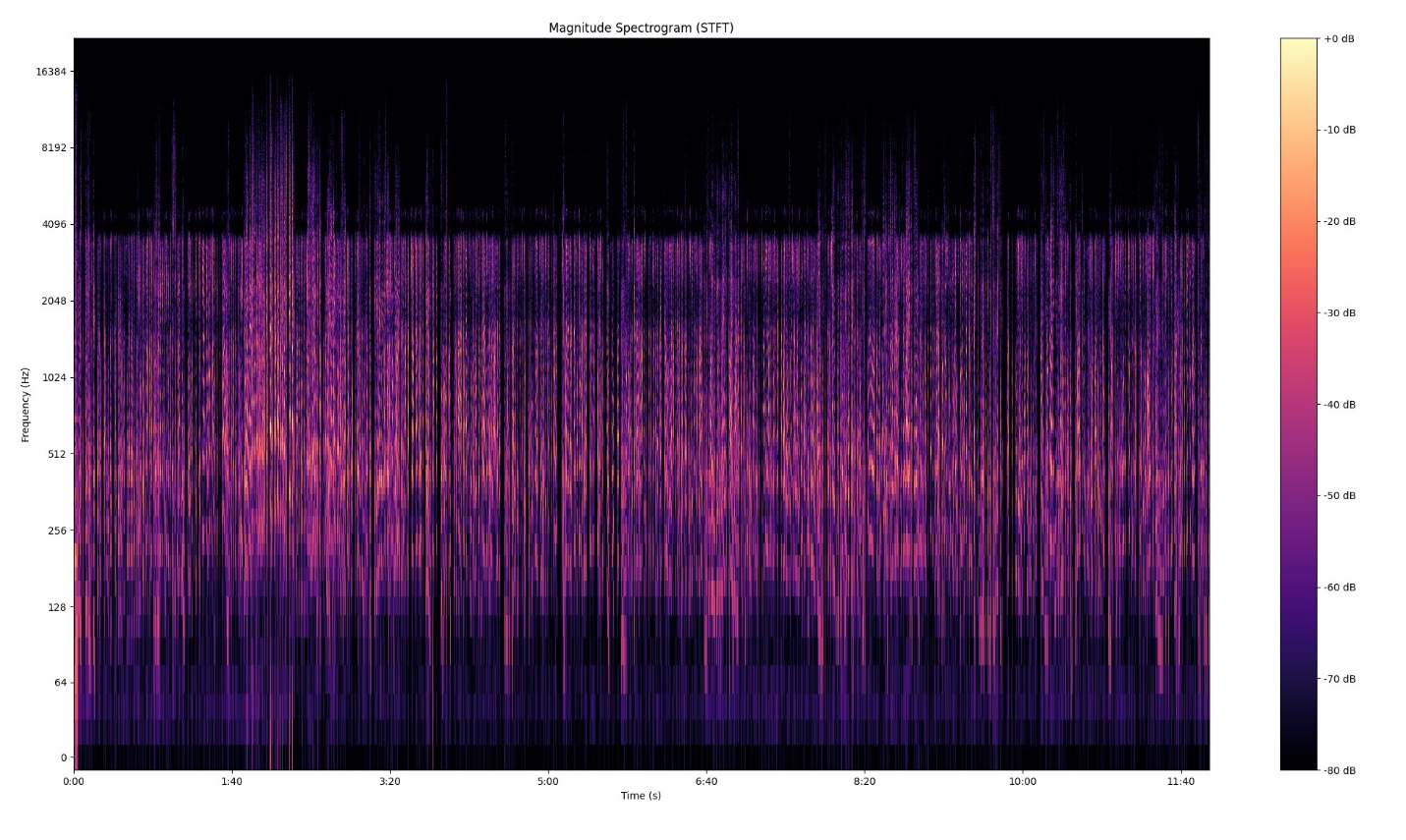
Intervalele rezultate sunt exportate într-un fișier .csv (time\_ranges2.csv)

Această tehnică este mai puțin rapidă ca prima, însă este potențial mai precisă. Într-adevăr, precedenta tehnică este oarecum limitată de fluctuațiile dezordonate ale semnalului. Prin aplicarea filtrului gaussian, semnalul devine mai puțin haotic.

# Abordarea nr. 3

Mai întâi, video-ul cu Gigi Becali a fost descărcat de pe YouTube în format audio (.mp3). Audio-ul a fost apoi citit folosind *librosa.load*.

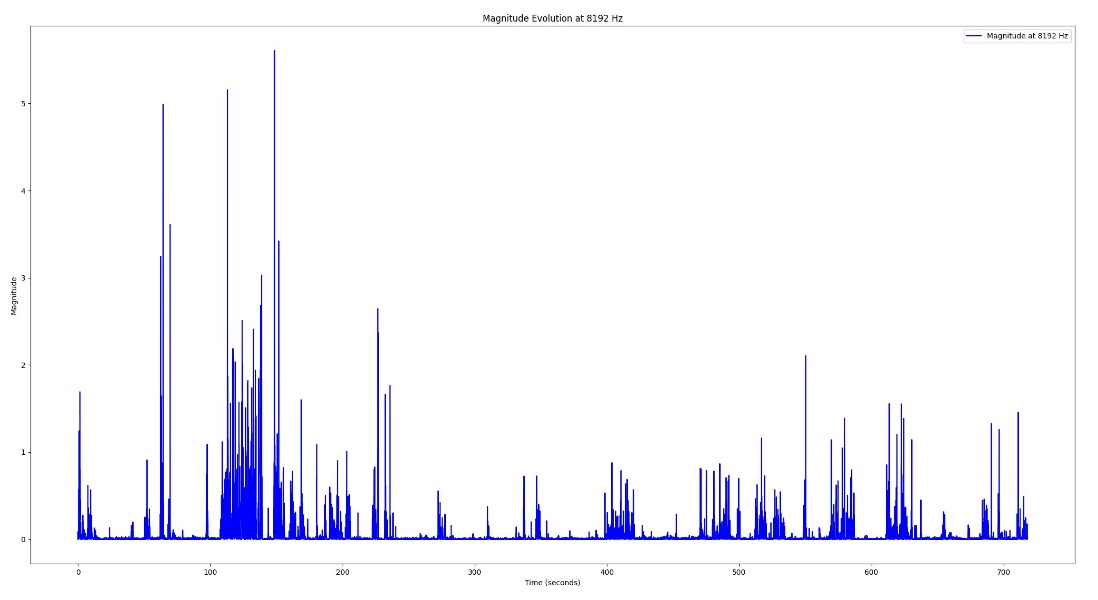
Folosind *librosa.stft(audio)*, am calculat transformata Fourier a semnalului. Aceasta este redată în graficul de pe pagina următoare.



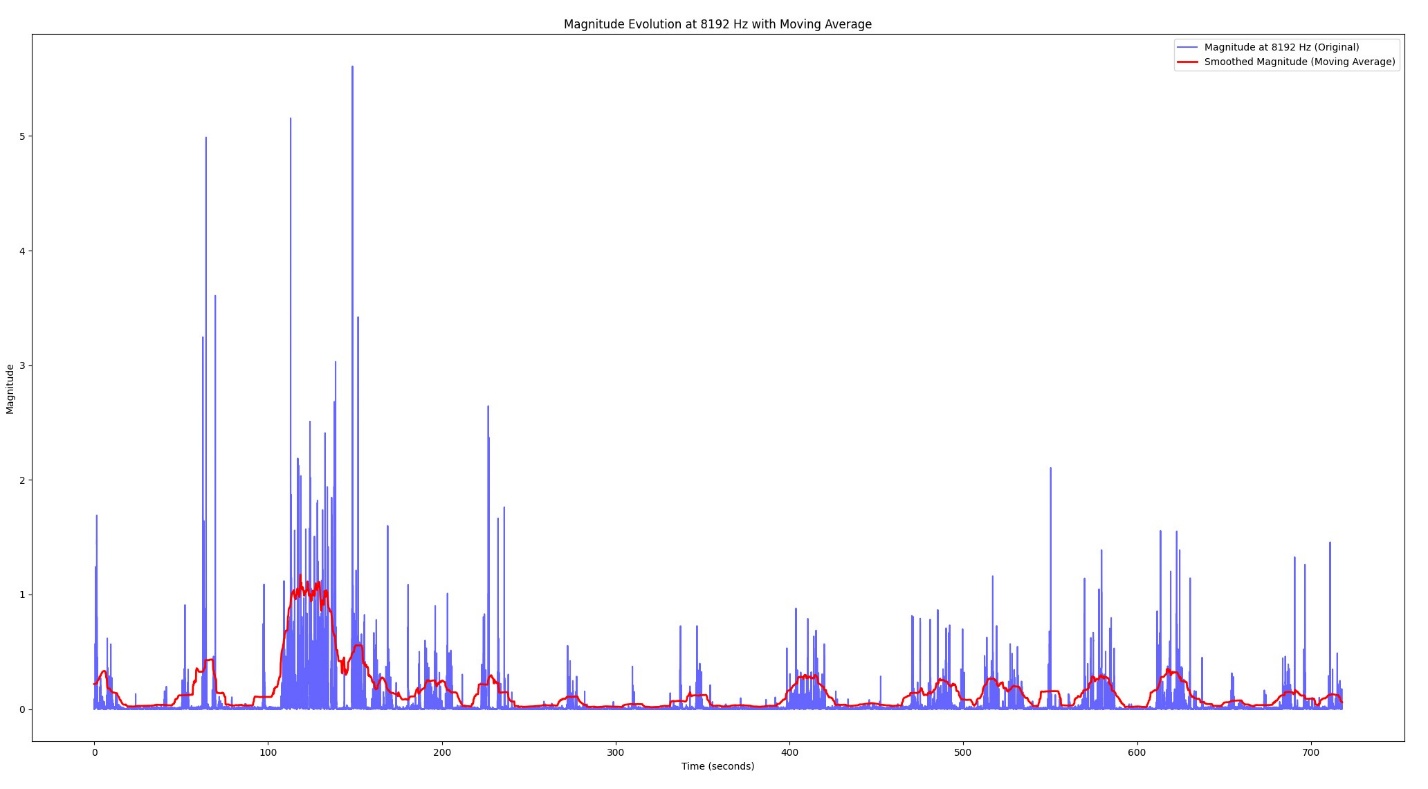
Pe orizontală avem timpul, iar pe verticală frecvența. Frecvențele sunt reprezentate pe o axă logaritmică, fiecare frecvență indicată pe axă fiind dublul precedentei. Codul de culori corespunde intensității semnalului. Intensitatea este exprimată în decibeli (dB), o unitate de asemenea logaritmică.

Observăm că frecvențele prezente în cea mai mare parte a fișierului audio sunt mai mici sau egale cu 4096 Hz. Într-adevăr, frecvențele înalte (8192 Hz sau peste) sunt preponderent absente.

Să luăm frecvența egală cu 8192 Hz și amplitudinea corespunzătoare (vezi graficul de mai jos).



Cu ajutorul unei medii mișcătoare (*moving average*), putem analiza evoluția amplitudinii semnalului corespunzător frecvenței de 8192 Hz.



Apoi, calculăm media funcției și alegem un prag, de exemplu triplul mediei. Dacă intensitatea depășește pragul, notăm intervalul ca „încins”. Raționamentul este următorul: atunci când strigă sau țipă, oamenii tind să emită sunete mai „ascuțite”, atingând astfel frecvențe mai înalte decât în mod obișnuit.

Această tehnică este „isteață”, însă poate deveni imprecisă în cazul unor voci care sunt în mod normal ascuțite, îndeosebi unele voci feminine.

# Concluzie

Codul de Python este disponibil pe platforma Github. De menționat că porțiuni din cod au fost generate cu ajutorul inteligenței artificiale. Este vorba despre părțile mai tehnice/matematice, cum ar fi mediile mișcătoare, învelitorile și transformatele Hilbert sau Fourier.

Așa cum precizam la început, nivelul meu de cunoaștere a limbajului Python este unul junior, nefiind familiarizat cu unele tehnici mai avansate. Cu toate acestea, m-am străduit să înțeleg codul scris.

Consider că rezultatele obținute sunt destul de precise. Mai mult decât atât, metodele alese sunt versatile, prin prezența unor parametri ajustabili.