

# Examen / Tarea

...

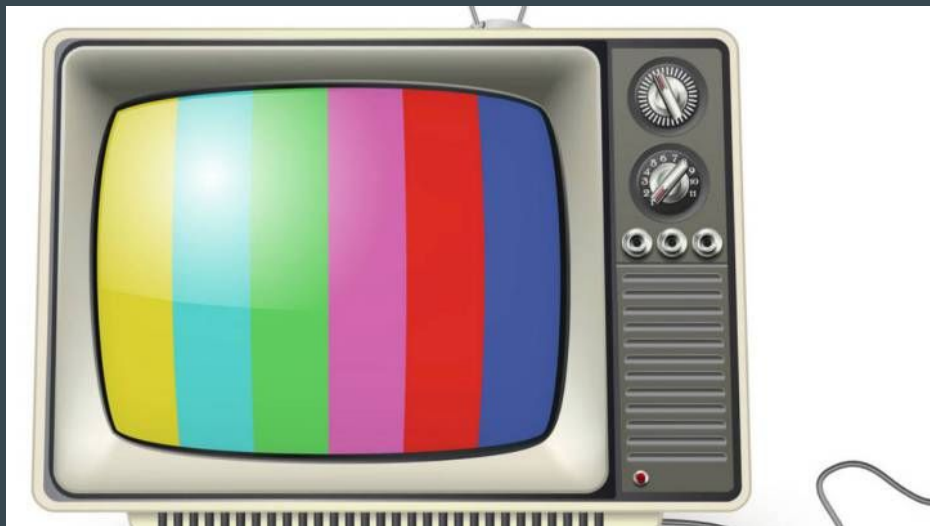
CC5213

Agosto, 2021

Ignacio Moore

# Motivación

- Tarea 2 incompleta
- Audio es algo más abstracto y así, mas complicado de comprender
- Ver que audios son los más populares en series, películas, videos, etc.



# Objetivo

- Crear un **buscador de audios repetidos** que actúe sobre cualquier material audio visual, como películas, series, televisión u otros audios.

## ¿Por que?

- Encontrar sonidos parecidos **difícil** para el oído humano.
- Es un área donde se puede **analizar y aprender** bastante.



# Entendiendo el problema

## Paso 1

Separar el audio de videos

## Paso 2

Crear descriptores de los audios

## Paso 3

Comparar descriptores y ver qué secuencia de descriptores son los más parecidos.

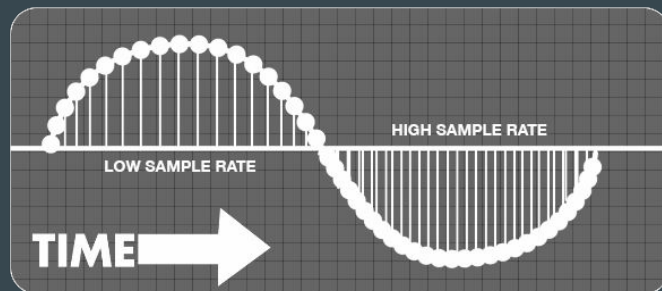
# Herramientas y Técnicas Utilizadas

## Paso 1

Separar el audio de los videos

### Herramientas y Técnicas:

- ffmpeg
- Sample Rate: 22 kHz
- .wav



# Herramientas y Técnicas Utilizadas

## Paso 2

### Descriptores

#### Herramientas y Técnicas:

- Librosa
- MFCC (Mel-frequency cepstral coefficients)
  - El mas comun
  - Reconoce voces
- Descriptores por segundo: 2
- Dimensión: 32



# Herramientas y Técnicas Utilizadas

## Paso 3

Buscador

### Herramientas y Técnicas:

- Numpy
- Scipy.spatial.distance.cdist
- Secuencias
- Pandas

fotograma	vecinos cercanos			
(417, 001)	(415, 001)	(410, 001)	(409, 001)	(421, 513)
(417, 002)	(409, 002)	(410, 002)	(420, 116)	(425, 089)
(417, 003)	(418, 222)	(415, 003)	(420, 113)	(410, 003)
(417, 004)	(420, 118)	(410, 004)	(415, 004)	(418, 223)
(417, 005)	(415, 005)	(410, 005)	(420, 119)	(407, 122)
(417, 006)	(410, 006)	(415, 006)	(420, 120)	(407, 123)
(417, 007)	(415, 007)	(410, 007)	(420, 121)	(418, 226)
(417, 008)	(410, 008)	(415, 008)	(420, 122)	(407, 125)
(417, 009)	(420, 123)	(412, 009)	(411, 009)	(415, 009)
(417, 010)	(410, 010)	(415, 010)	(412, 010)	(420, 124)



# Dataset

- Tarea 2
- Transmisión de televisión de MEGA
- Comerciales ya identificados
- Buena base para hacer un **análisis de precisión**



SAMSUNG

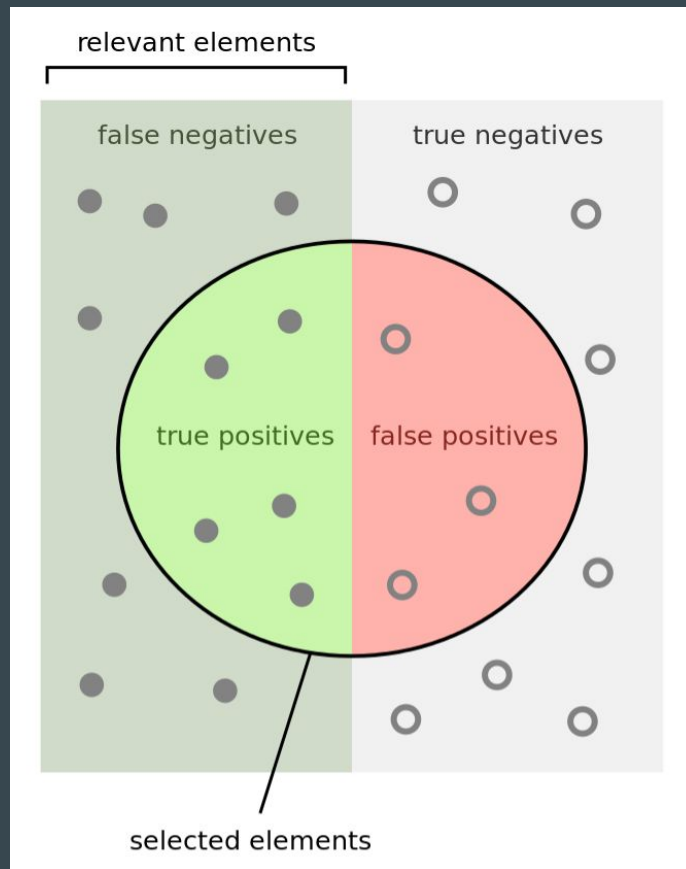




# Analysis

## Que se busca medir

- Variación de cantidad de vecinos
- Variación de longitud de secuencia
- Precision



# Demo

...

Python 3  
Windows 10

# Resultados y Conclusiones

- Es más fácil encontrar una duplicación de secuencias cortas de audios.
- Al tener mas vecinos, no necesariamente mejora los resultados.
- Los resultados no son tan buenos, alcanzando un máximo de aproximadamente 2% de precisión.
- Código fuente y README en [https://github.com/ignaciomoore/rim\\_project](https://github.com/ignaciomoore/rim_project)