# Identificador de voz con coeficientes MEL

Dayana Gonzalez Vargas
*dayana.gonzalez@urosario.edu.co*
*Estudiante de matemáticas aplicadas y ciencia de la computación.*

David Santiago Flórez Alsina
*davidsa.florez@urosario.edu.co*
*Estudiante de matemáticas aplicadas y ciencia de la computación.*

(Universidad del Rosario)
(Dated: October 11, 2021)

## Introducción

En la actualidad con los avances tecnológicos hemos llegado al punto en el que es posible tecnológicamente identificar a la persona hablante para diversos propósitos entre ellos asuntos de seguridad, recibir y ejecutar comandos por voz, de estas anteriores nos centraremos en poder identificar el usuario hablante mediante la manipulación de la señal de voz utilizando los coeficientes mel-cepstrum y realizando una clasificación con regresión logística multinomial.

## Metodología

Para el desarrollo del identificador de voz, entrenamos un modelo de regresión logístico votador con tres categorías *(1-Dave, 2-Dayana, 3-Otro)* basándonos en un dataset construido mayoritariamente por nosotros, el cual se compone de catorce muestras de audio de Dayana, catorce de Dave y catorce de personas desconocidas para el modelo, adicionalmente con las pruebas nos dimos cuenta que era conveniente añadir unas señales de ruido *(añadimos cuatro)*, para poder clasificar más claramente con el modelo.

Respecto a las muestras de audio utilizadas en el entrenamiento estas fueron grabadas en diversos dispositivos y por ello tienen diversas frecuencias de grabación, las frecuencias manejadas son: $12.5kHz$, $16kHz$, $44.1kHz$, $48kHz$. Por defecto la aplicación maneja los siguientes parámetros para tomar la señal de audio: Frecuencia de muestreo de 16 $kHz$, 24 bits de resolución y un solo canal de grabación, algunos audios se grabaron en computador directamente con estos parámetros, sin embargo, estos parámetros se usan principalmente en la app para realizar el testeo en el modelo de regresión logística multinomial.

En cuanto a las muestras de audio de testeo estas también fueron tomadas desde distintos dispositivos y se presentan en el mismas rango de frecuencias que en el caso de los audios de entrenamiento, al igual que en el caso del entrenamiento también se usaron grabaciones hechas con los parámetros de la app para testear, la cantidad de muestras de audio de testeo son veinticuatro, con ocho muestras para cada categoría.

Para entrenar y testear concretamente el modelo de regresión logística multinomial se tienen los siguientes procesos que sigue los respectivos algoritmos:

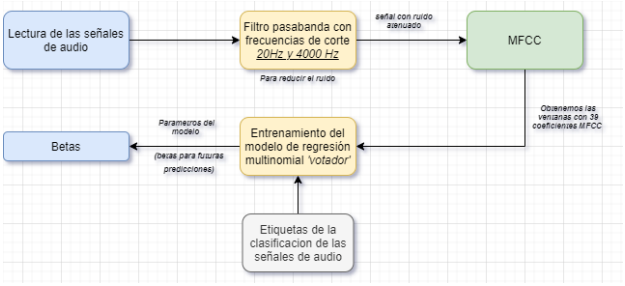


FIG. 1: *Proceso del algoritmo de entrenamiento, el concepto de modelo de regresión logística multinomial votador lo abordaremos en la sección teórica.*

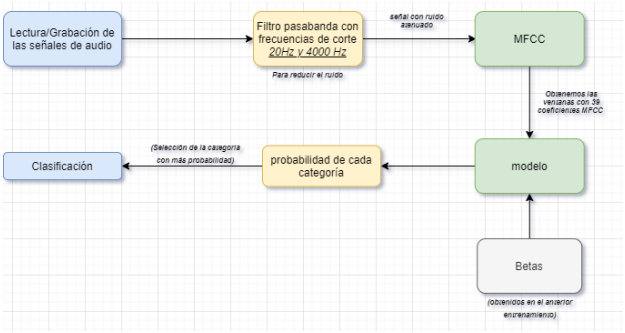


FIG. 2: *Paso a paso del algoritmo de testeo*

Finalmente con lo anterior nos dispusimos a plantear la estructura de la interfaz gráfica y su funcionamiento.

## Conceptos y Teoría

### A. La señal de voz y la audición

Para empezar teniendo en cuenta el proceso de generación de la voz ilustrado aquí:

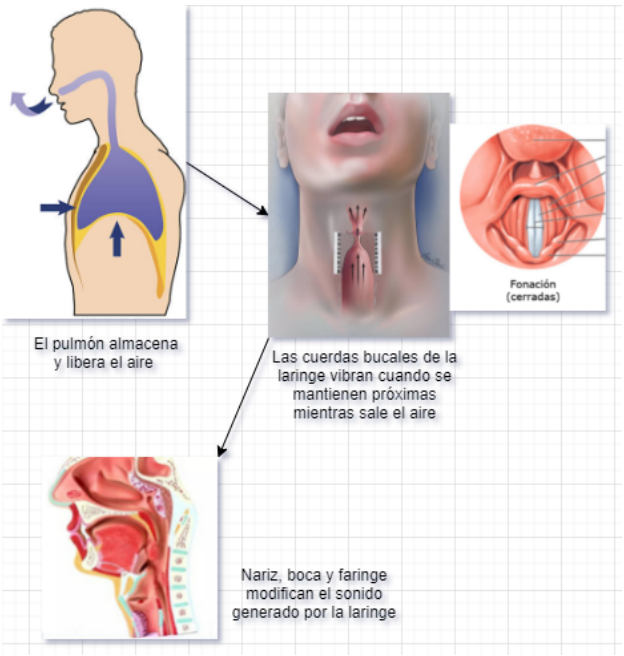


FIG. 3: *Proceso de fonación*

Se reconoce que la voz es la convolución entre los pulsos generados por la glotis (cuerdas bucales) y el filtro que hace el tracto bucal, matemáticamente esto tendrá su utilidad como veremos más tarde.

También es útil saber que el rango de frecuencias sobre el cual se ubica la voz humana está entre $20Hz$ y $4kHz$ por lo que podremos usar un pasa banda con frecuencia de corte en estos 2 puntos y limpiar parcialmente nuestra señal de algunos ruidos.

Además, las voces de los hombres y las mujeres tienen diferencias notables en como son sus valores de frecuencia, teniendo que la voz de los hombres esta en un rango de frecuencia fundamental de $85Hz$ a $180Hz$ y el rango de la voz de una mujer se encuentra entre las frecuencias de $165Hz$ a $255Hz$, por lo que los tractos bucales de los hombres poseen frecuencias más bajas que la de las mujeres.

Ahora, también hay que mencionar que nuestra forma de oír tiene una escala logarítmica, es decir que escuchamos mejor las frecuencias bajas, por ejemplo si tenemos una señal en frecuencia de $100Hz$ y otra en $200Hz$ sentiremos mejor la diferencia entre ambos sonidos en comparación con otros dos sonidos en $1000Hz$ y $1100Hz$, esto porque la primera pareja está en frecuencias más bajas.

La anterior relación la podemos ver a través de la siguiente fórmula con su gráfica:

$$m = (1127.0148) \cdot \log(1 + \frac{f}{700})$$

*Para convertir de f (frecuencias [Hz]) a m (frecuencias Mel [mels])*

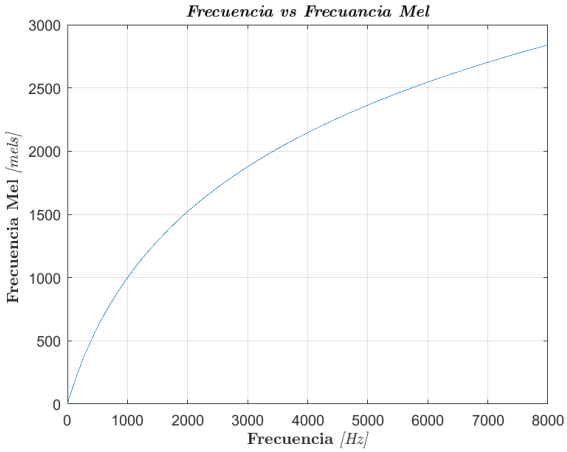


FIG. 4: *Comparación entre los valores de frecuencia normales de la señal y la escala de frecuencias Mel.*

### B. MFCC

Los coeficientes cepstrales de frecuencias Mel *(MFCC)* son comúnmente utilizados en el reconocimiento de voz debido a su capacidad de poder separa la señal en dos componentes que son; el tracto bucal y las frecuencias producidas por las cuerdas bucales, mientras a su vez se pone en la escala del oído humano *(La idea aquí es poder hacer replicación de como el humano percibe el sonido para después usar esta forma de percibir para crear clasificadores o entender audio)*, entonces para encontrar estos coeficientes se utiliza el siguiente algoritmo:

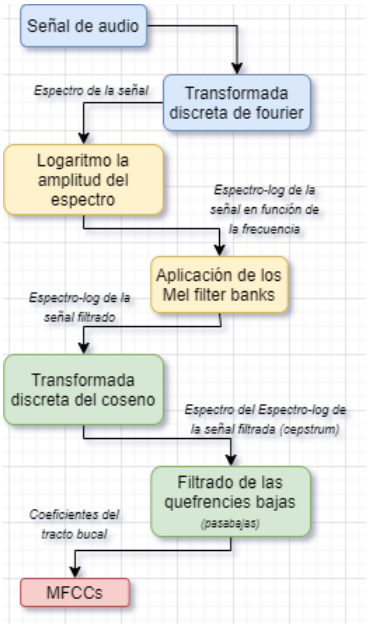


FIG. 5: *Proceso de obtención de los MFCC.*

Debido a que la señal de audio es un convolución podemos tener que; $x(t)$, representa la voz de una persona, $e(t)$, represente las frecuencias producidas por las cuerdas bucales y $h(t)$, representa el tracto bucal, sabemos que la voz esta constituida por estas dos componentes anteriormente mencionadas.

$$x(t) = e(t) \cdot h(t)$$

Al obtener la transformada discreta de Fourier y debido a que la señal es una convolución obtenemos:

$$X(j\omega) = E(j\omega) \cdot H(j\omega)$$

Con esto podemos aplicar logaritmo y convenientemente tenemos:

$$log(X(j\omega)) = log(E(j\omega)) + log(H(j\omega))$$

Ahora, se aplica el banco de filtros mel para generar esta sensación de mejor resolución en las bajas frecuencias y en escala logarítmica.

$$S(j\omega) = log(X(j\omega)) \cdot melFilterBank(j\omega)$$

Tras esto podemos tratar esta nueva señal $S(j\omega)$ como una señal normal y aplicarle la transformada discreta del coseno para obtener entonces las *'frecuencias'* de esta señal. Dado que las frecuencias del tracto vocal eran bajas en comparación con las de la garganta, podemos aplicar un pasa-bajas y quedarnos con las frecuencias bajas, que son las que nos interesan para caracterizar una persona, de allí salen los MFCC's.

Todo el procedimiento anteriormente explicado, se realiza en nuestra aplicación mediante la función `mfcc` de `matlab`, donde nosotros decidimos tomar treinta y ocho coeficientes para que se pudiera obtener una mayor información de cada uno de los segmentos o ventanas que se generan de la señal, donde esto contribuya a una mejor clasificación en nuestro modelo de regresión logística multinomial.

### C. Modelo regresión logístico multinomial votador

Realizamos un modelo de regresión logística para poder clasificar una señal de audio como una de las tres categorías *(1-Dave, 2-Dayana, 3-Otro)*, en el cual se realizaron los anteriores algoritmos dados en la metodología del proyecto.

*Recordemos que la formula a seguir de una regresión logística:*

$$P(\pi_i) = \log(\frac{\pi_i}{1-\pi_i}) = \beta_0 + \beta_1 X_{1,i} + ..... + \beta_{p,i} X_{p,i}$$

*donde, i nos indica la categoría a la que se refiere y p el número de descriptores.*

En este modelo realizado tomamos los descriptores como el número de coeficientes dados, debido a que la función `mfcc` de `matlab` nos genera una matriz de tamaño(ventanas generadas $\times$ el número de coeficientes). Como la función MFCC divide nuestra señal de audio en segmentos del mismo tamaño, donde en cada uno de ellos se genera la ventana y se obtienen sus coeficientes, lo que queremos ver es como esos valores dados en cada una de esas ventanas contribuyen a una probabilidad de que una persona sea clasificada en las respectiva categoría.

Mediante el entrenamiento del modelo con nuestras muestras de entrenamiento y etiquetas ya conocidas, se obtiene los coeficientes betas, los cuales nos determinan la contribución de cada una de las variables descriptoras en la predicción. Estos betas son utilizados para realizar el testeo y generar las debidas probabilidades de cada categoría en cada una de las ventanas, es decir, en cada uno de los segmentos de la señal, entonces se saca media de estas probabilidades de la respectiva categoría y obtenemos el máximo de estos, el cual nos indica cual es la categoría a la cual este audio de testeo pertenece.

### D. Potenciales aplicaciones *(distintas de la que estamos tratando)*

Entre las posibles aplicaciones que tienen los MFCC's están:

- ***Sistema de reconocimiento de palabras***, para este sistema haríamos un modelo que pase de voz a texto y que a partir del texto generado identifique si esta palabra hace parte de un diccionario de palabras para el lenguaje que se está hablando, el diagrama que describiría este proceso a más detalle es:

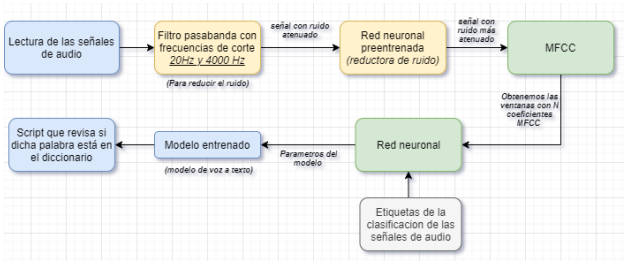


FIG. 6: Diagrama del algoritmo para el reconcomiendo de palabras

- ***Sistema de Karaoke***, para un sistema de karaoke tendríamos en cuenta el audio que genera la persona que sigue el karaoke y la canción a seguir, primero con un reconocimiento de palabras podemos leer las palabras de dicha canción, luego realizando un filtro para captar mejor la voz de la persona que se encuentra entre $20Hz - 4kHz$, realizamos una red neuronal para primero reducir la cantidad de ruido que se encuentre en el audio, también se pueden realizar diferentes filtros como un pasa-bandas teniendo en cuenta esos puntos de corte, luego encontramos los coeficientes de mel-cepstrum, los cuales serán utilizados para el testeo en un modelo de clasificación binario, el cual nos pueda decir si esa palabra que dijo la persona se encuentra o no en la canción, este algoritmo también será entrenado con el script que sale del algoritmo de reconocimiento de palabras.

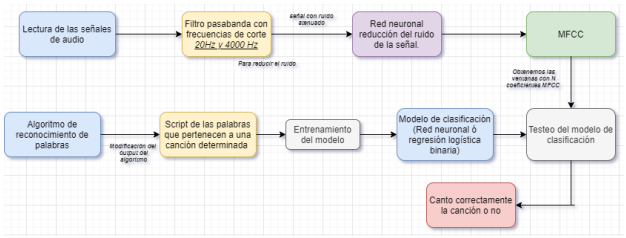


FIG. 7: Diagrama del algoritmo para un karaoke

## Análisis de resultados y discusión

Con lo explicado anteriormente desarrollamos la siguiente interfaz:

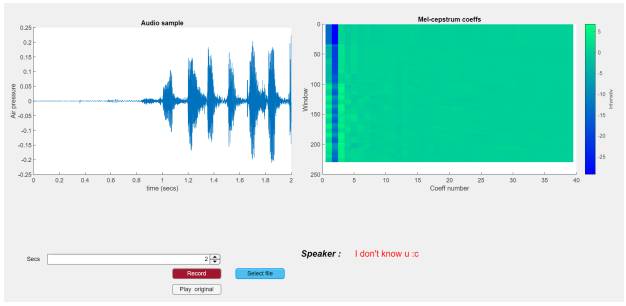


FIG. 8: *Interfaz gráfica implementada*

Nuestra interfaz posee dos gráficas, la primera se refiere al la señal de audio original y la segunda nos demuestra los coeficientes mel obtenidos en cada una de las ventanas y como es la intensidad de estos, recordando que se obtienen los 38 coeficientes de cada ventana con las bajas frecuencias. Comparando las gráficas dadas por hombres y mujeres se puede notar que los hombres tienen frecuencias mucho más bajas que las de las mujeres donde el cepstrum de los hombres toma valores muchos más grandes que los valores que se toman en el cepstrum de las mujeres.

Al evaluar el modelo de regresión por votación con todos los audios de testeo anteriormente mencionados obtuvimos los siguientes resultados condensados en esta matriz de confusión:

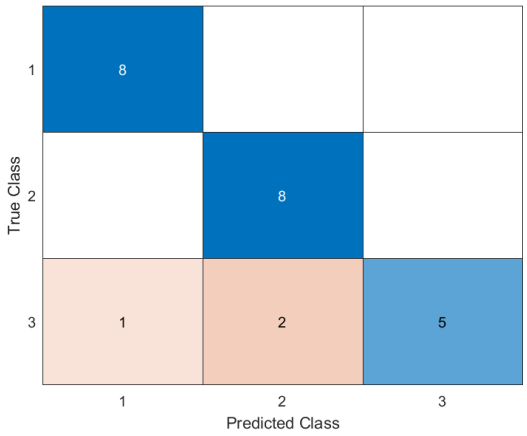


FIG. 9: *Matriz de confusión del modelo con todos los audios de testeo, con tres categorías (1-Dave, 2-Dayana, 3-Otro).*

Durante el proceso de testeo que realizamos, nos dimos cuenta, que por ejemplo, la computador donde estaba grabando uno de los integrantes del equipo era bastante ruidoso, el algoritmo aprendió a distinguirlo entre otras cosas por el ruido de su ordenador a diferencia de otros audios, es por ello que introducimos algunas muestras de audio con ruido y además incluimos muestras de ruido exclusivamente, los cuales etiquetamos como persona desconocidas, con esto el problema del ruido se solventó para las veces que probamos.

Además, vimos que con muestras de audio algo más largas que un segundo, las predicciones son mejores, por esto es que pusimos como mínimo tiempo de duración de la grabación 2 segundos en la app.

Con respecto a las condiciones bajo las cuales el sistema podría fallar, están las situaciones en donde el ruido que no fue filtrado se parezca al ruido de alguno de los usuarios, aunque ya se entrenó al modelo con datos que buscan corregir esto, este caso aún puede ocurrir. Además, podría darse el caso en que otra persona haga pausas similares a las que realizan los usuarios principales *(1-Dave, 2-Dayana)* y el sistema podría fallar allí. Por otra parte el audio de prueba dado aún tiene más características que se pueden extraer y que pueden mejorar la calidad

de predicciones del sistema, por ejemplo en varios papers y repositorios vistos se utiliza el estimado de la frecuencia fundamental del audio *(con la función pitch de matlab)* para alimentar el clasificador.

Con lo anterior creemos que para realizar un mejor trabajo en el pre-procesamiento de los datos se deben atenuar o eliminar factores externos a la voz de la persona, por ejemplo en diversos parpers vistos se utilizan algoritmos para extraer solo las partes en que la persona habla en un audio, de esta forma podría evitarse que el algoritmo aprenda a distinguir entre otras cosas una persona de otra por las pausas que realiza al hablar.

También aplicar una red neuronal pre-entrenada para la tarea de borrar el ruido del audio o atenuarlo lo más posible sería de gran ayuda, ya que le permite a la función MFCC trabajar directamente con una señal más pura y así el algoritmo aprendería mejor.

---

## Bibliografía


[1] Sánchez. (2016, 2 septiembre). *Extracción de parámetros para reconocimiento de voz– Rubén Sánchez.* Blog Rubén Sánchez. http://rubensm.com/extraccion-de-parametros-para-reconocimiento-de-voz/

[2] Ruiz, I. V. M. (2013, 28 marzo). *MFCCs.* Ivan Vladimir Meza Ruiz. https://turing.iimas.unam.mx/7Eivanvladimir/posts/mfcc/

[3] Valerio Velardo - The Sound of AI. (2020, 5 octubre). *Mel-Frequency Cepstral Coefficients Explained Easily* [Vídeo]. YouTube. https://www.youtube.com/watch?v=4_SH2nfbQZ8

[4] colaboradores de Wikipedia. (2019, 22 diciembre). *Escala Mel.* Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Escala_Mel

[5] Luis Gabriel Toro Cerón. (2018). *Análisis de Estrés en la Voz Utilizando Coeficientes Cepstrales de Mel y Máquina de Vectores de Soporte.* http://45.5.172.45/bitstream/10819/6041/1/Analisis_Estres_Voz_Toro_2018.pdf

[6] R.J,Y.T,H.F,G.M,M.A I.A. (2021). *Expert Systems with Applications* (Vol. 171). Elsevier. https://www.sciencedirect.com/science/article/abs/pii/S0957417421000324?viaDihub

[7] D. Anggraeni. (2018). *The Implementation of Speech Recognition using Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machine (SVM) method based on Python to Control Robot Arm.* PAPER OPEN ACCESS.

[8] Sloveby Suksri. (2015, septiembre). *Speech Recognition using MFCC.* ResearchGate. https://www.researchgate.net/publication/281446199_Speech_Recognition_using_MFCC

[9] Jon Gudnason, Mike Brookes. (2008). *VOICE SOURCE CEPSTRUM COEFFICIENTS FOR SPEAKER IDENTIFICATION.* Exhibition Road.

[10] Gulmira K., Oleg N., Valery V., Kasymbeg M. (2017). *Pre-processing Voice Signals for Voice Recognition Systems. Penza state university.*