UNIVERSIDAD NACIONAL MICAELA BASTIDAS DE APURÍMAC

FACULTAD DE INGENIERÍA

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA INFORMÁTICA Y SISTEMAS

TESIS

Determinación del mejor método de clasificación para el desarrollo de software de identificación de voz en los docentes de la UNAMBA - 2018

Presentado por:

Betsabe Milagros Ccolqque Ruiz

Para optar el Título de Ingeniero Informático y Sistemas

Abancay, Perú

2021

UNIVERSIDAD NACIONAL MICAELA BASTIDAS DE APURÍMAC

FACULTAD DE INGENIERÍA

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA INFORMÁTICA Y SISTEMAS

TESIS

DETERMINACIÓN DEL MEJOR MÉTODO DE CLASIFICACIÓN PARA EL DESARROLLO DE SOFTWARE DE IDENTIFICACIÓN DE VOZ EN LOS DOCENTES DE LA UNAMBA - 2018

Presentado por **Ccolqque Ruiz Betsabe Milagros**, para optar el Título de:

INGENIERO INFORMÁTICO Y SISTEMAS

Sustentado y aprobado el 20 de octubre del 2021 ante el jurado evaluador:

Presidente: _____
Mag. Francisco Cari Incahuanaco

Primer Miembro: _____
D.Sc. Erech Ordoñez Ramos

Segundo Miembro: _____
Mag. Evelyn Naida Luque Ochoa

Asesor (es) : _____
D.Sc. Ecler Mamani Vilca

Agradecimiento

Quiero agradecer a Dios, porque él sabe cómo tener detalles conmigo y a mis padres por su paciencia, comprensión y amor reflejado en insistencia. A mi asesor por la guía, ánimo e impulso, a mis docentes por su contribución en mi formacion profesional y a mí misma, por hacer posible este logro y, en general, a todos los que contribuyeron en la realización de esta investigación. ¡GRACIAS!

Dedicatoria

A Dios, por su amor y misericordia.

A mis padres, por su interminable paciencia y comprensión.

Y a mí, porque lo logré.

“DETERMINACIÓN DEL MEJOR MÉTODO DE CLASIFICACIÓN PARA EL DESARROLLO DE SOFTWARE DE IDENTIFICACIÓN DE VOZ EN LOS DOCENTES DE LA UNAMBA - 2018”

Ingeniería Informática, Industria y Sociedad

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

En los últimos años se ha producido un crecimiento exponencial de las tecnologías de la información debido a la automatización de procesos, almacenamiento de la información y el uso del internet; de esta manera se ha incrementado la interacción humano-computador utilizando la voz, lo cual nos otorga más seguridad en la medida que esté bien implementada. La identificación de un locutor no va a depender únicamente de como éste sea o que lleve puesto, sino que se basará en saber quién es, mediante la emisión de la señal de su voz. Esto conlleva a la necesidad de almacenar grandes cantidades de información para luego ser analizados y a realizar la debida clasificación que describa sus principales características, la que se puede dar por su similitud o diferencia.

La investigación se ubica dentro del ámbito de la inteligencia artificial, específicamente en el procesamiento del lenguaje natural, utilizando la estrategia estadística en el estudio de ciertos métodos de agrupamiento de datos para la identificación de voz. Por lo tanto se busca determinar el mejor método para el desarrollo de software de identificación de voz, entre los siguientes: Los Modelos Ocultos de Markov, los Modelos de Mezclas Gaussianas y los Máquinas de Vectores Soporte.

RESUMEN

El procesamiento del lenguaje natural es una rama de la inteligencia artificial, que actualmente se aplica en todas las áreas, como en el reconocimiento, identificación, escritura y traducción de voz mediante el uso de algoritmos y modelos de clasificación que aprenden a partir de los datos en volúmenes grandes o medianos a través de la técnica estadística y del aprendizaje basado en máquina. Al existir una variedad de métodos utilizados para la identificación de voz, actualmente hay pocos estudios que sugieran la confiabilidad del mejor método generando desconcierto entre los desarrolladores de software para la identificación de voz.

Esta investigación tiene como propósito determinar el mejor método entre los tres investigados, los Modelos Ocultos de Markov, los Modelos de Mezclas Gaussianas y las Máquinas de Vectores Soporte, para el desarrollo de software de identificación de voz. Con esta finalidad hemos utilizado un diseño metodológico cuantitativo, aplicando instrumentos como tablas de comparación para el registro de cada método, a 32 docentes de la UNAMBA sede central. Nosotros efectuamos la evaluación mediante la métrica de la Accuracy y el estadístico denominado “análisis de varianza de un factor: diseño por bloques aleatorizados”.

Nuestros resultados muestran un 95.83% de Accuracy para los Modelos de Mezclas Gaussianas, un 94.79% de Accuracy para los Modelos Ocultos de Markov y en último lugar un 30.21% de Accuracy para las Máquinas de Vectores Soporte.

Las simulaciones que hemos realizado muestran que el método más efectivo en la identificación de voz son los Modelos de Mezclas Gaussianas que sobresalen de sus similares: los Modelos Ocultos de Markov, que ha obtenido el segundo lugar de Accuracy y las Máquinas de Vectores Soporte que es un método supervisado que presenta desventajas en su implementación. Esta investigación busca ser un aporte en la decisión del método a utilizar para el desarrollo de software de identificación de voz. Finalmente recomendamos estudios sobre las métricas de evaluación para modelos de clasificación aplicados a la identificación de voz y la forma de etiquetado de datos para voz en las Máquinas de Vectores Soporte.

Palabras clave: *Modelos de Mezclas Gaussianas, Modelos Ocultos de Markov, Máquinas de Vectores Soporte, identificación y procesamiento del lenguaje natural.*

ABSTRACT

Natural language processing is a branch of artificial intelligence, which is currently applied in all areas, such as speech recognition, identification, writing and translation through the use of algorithms and classification models that learn from data in large or medium volumes through statistical technique and machine-based learning. As there are a variety of methods used for voice identification, there are currently few studies that suggest the reliability of the best method, causing confusion among developers of voice identification software.

The purpose of this research is to determine the best method between the three investigated, the Hidden Markov Models, the Gaussian Mixture Models and the Support Vector Machines, for the development of voice identification software. For this purpose, we have used a quantitative methodological design, applying instruments such as comparison tables to record each method, to 32 teachers at UNAMBA headquarters. We carried out the evaluation using the Accuracy metric and the statistic called "one-factor analysis of variance: randomized block design".

Our results show a 95.83% accuracy for the Gaussian Mixture Models, a 94.79% accuracy for the Hidden Markov Models and lastly a 30.21% accuracy for the Support Vector Machines.

The simulations that we have carried out show that the most effective method in voice identification are the Gaussian Mixture Models that stand out from their similar ones: the Hidden Markov Models, which has obtained the second place of Accuracy and the Support Vector Machines that is a supervised method that has disadvantages in its implementation. This research seeks to be a contribution in the decision of the method to be used for the development of voice identification software. Finally, we recommend studies on evaluation metrics for classification models applied to voice identification and the form of data labeling for voice in Support Vector Machines.

Keywords: *Gaussian Mixture Models, Hidden Markov Models, Support Vector Machines, identification and natural language processing.*

CAPÍTULO I
PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción del problema

Actualmente dentro del campo de procesamiento del lenguaje natural se realizan estudios que intenta replicar la facultad de lenguaje humano como el reconocimiento, identificación, verificación, escritura y traducción de voz, a través de algoritmos, métodos y técnicas con los cuales se desarrollan herramientas aplicadas a los diferentes casos de la vida real.

Una de la técnicas de identificación es la de voz, actualmente esta técnica es más portable, puesto que no es necesario recordar los conjuntos de caracteres que se ingresan a los formularios que conocemos como password. Sin embargo actualmente existen bastantes métodos para identificar la voz, según Sayed Jaafer Abdallah (2012) indica que la mayoría de los métodos de identificación de voz, presentan inconvenientes durante su desarrollo, al momento de clasificar los vectores de características a causa de la configuración de los parámetros y en el desajuste entre las condiciones de entrenamiento y prueba, entonces si queremos mejorar la robustez del software de identificación de voz, necesitaremos que los parámetros estén bien configurados para que la extracción sea eficiente, de esta manera los resultados serán los deseados.

Actualmente existen muchos métodos para la identificación de voz, sin embargo también existen muchas dudas respecto a qué método utilizar para implementar un software de identificación de voz, cuál modelo matemático a aplicar es el mejor o más adecuado, entonces se producen desaciertos en el desarrollo de software; si se comercializa algún producto con errores o inestable dado que los componentes para la captura y el procesamiento suelen ser de costo elevado, implica que una errada decisión de compra y venta de estos sistemas biométricos conlleve una pérdida económica al comprador o que su implementación sea costosa y poco efectiva. En este tipo de sistemas biométricos los datos respecto al porcentaje de acierto o efectividad al identificar un número de locutores es de gran importancia, porque de ello depende la tasa de éxito o la tasa de fracaso que calificará al software de

identificación de voz, lo que determinará en la decisión de compra del usuario que tiene la intención de adquirirlo. El usuario necesita confiar en la calidad del sistema biométrico de identificación de voz, lo cual es de mucha importancia debido a que son sistemas de seguridad y la mala experiencia del usuario produciría percepciones negativas. Así, esta investigación ayudará al desarrollador de software a reducir el tiempo de búsqueda y prueba para determinar un método de clasificación exitoso para la identificación de voz.

Por ello se estudió ciertos métodos que serán utiles en el desarrollo de software de identificación de voz. Según Ochoa y otros (2008) en "**Identificación biométrica de locutores para el ámbito forense: estado del arte"**, sugieren la utilización de estos tres métodos HMM, GMM y el SVM, y de estos tres no hay mención de cuál de ellos es el mejor, por lo que se plantea determinar cuál es el mejor método para el desarrollo de software de identificación de voz.

1.2 Enunciado del problema

1.2.1 Problema general

¿Cuál es el mejor método entre los Modelos Ocultos de Markov, los Modelos de Mezclas Gaussianas y las Máquinas de Vectores Soporte, para el desarrollo de software de identificación de voz en los docentes de la UNAMBA - 2018?

1.3 Justificación de la investigación

Esta investigación busca ayudar a los desarrolladores de software orientados a la identificación de voz, brindando una información sobre el método mas eficiente que podrían aplicar en el desarrollo de un software de identificación de voz, el cual generará beneficios como: el ahorro de tiempo, ya que la búsqueda de un modelo que identifique correctamente la voz toma tiempo; la garantía de calidad, para evitar la mala experiencia del usuario se obtiene del accuracy y, finalmente, la reducción de gastos en el desarrollo y la comercialización de software.

Con el avance de la tecnología, una de las barreras que enfrenta el mundo moderno es la dificultad para adaptarse a las nuevas tecnologías, existiendo múltiples aplicaciones para diversas áreas, entre ellas la "identificación y autentificación", que se presenta desde el control de acceso en celulares móviles, hasta los sistemas de seguridad en línea telefónica, encontrándose estas nuevas tecnologías con una interacción humano-computador al alcance de todos.

Ya que una de las dificultades que presenta el desarrollo de un software de identificación de voz son los múltiples métodos que existen, lo que puede conducirnos a una mala elección del método a utilizar en la identificación de voz, por ello con la intención de prevenir esta mala elección se realiza el presente estudio basado en una comparación entre 3 métodos con el fin de determinar el método que nos entregue los mejores resultados.

CAPÍTULO II
OBJETIVOS E HIPÓTESIS

2.1 Objetivos de la investigación

2.1.1 Objetivo general

Determinar el mejor método entre los Modelos Ocultos de Markov, los Modelos de Mezclas Gaussianas y las Máquinas de Vectores Soporte para el desarrollo de software de identificación de voz en los docentes de la UNAMBA.

2.1.2 Objetivos específicos

- Analizar e implementar el conjunto de datos de docentes.
- Analizar y ejecutar el método Modelos Mezclas Gaussianas (GMM).
- Analizar y ejecutar el método Modelos Ocultos de Markov (HMM).
- Analizar y ejecutar el método Máquinas de Vectores Soporte (SVM).
- Determinar el mejor método mediante métricas de evaluación para modelos de clasificación.

2.2 Hipótesis de la investigación

2.2.1 Hipótesis general

Al clasificar correctamente los métodos: Los Modelos Ocultos de Markov, los Modelos de Mezclas Gaussianas y las Máquinas de Vectores Soporte, entonces se determina el mejor método para el desarrollo de software de identificación de voz en los docentes de la UNAMBA.

2.3 Operacionalización de variables

Tabla 1 — Operacionalización de variables

VARIABLE	DIMENSIONES	INDICADORES	ÍNDICE
VI: VOZ DE LOS DOCENTES Es la señal emitida por los docentes.	**Grabación de la voz de los docentes**	Parámetros de calidad de los audios de los docentes	- **Tasa de muestreo**
			- **Resolución de bits**
			- **PCM**
VD: MÉTODO DE CLASIFICACIÓN Son métodos utilizados en la clasificación de patrones.	**Modelos Ocultos de Markov**	Accuracy	- **Valor porcentual**
	Modelos de Mezclas Gaussianas	Accuracy	- **Valor porcentual**
	Máquinas de Vectores Soporte	Accuracy	- **Valor porcentual**

CAPÍTULO III
MARCO TEÓRICO REFERENCIAL

3.1 Antecedentes

3.1.1 Antecedentes internacionales

- **SM Kamruzzaman, ANM Rezaul Karim, Md. Saiful Islam y Md. Emdadul Haque, "Speaker Identification using MFCC-Domain Support Vector Machine", University of Rajshahi, Bangladesh, 2010**. Se trata de una revisión bibliográfica sobre una técnica de identificación del hablante dependiente de texto utilizando las Máquinas de Vectores de Soporte de Dominio y MFCC, donde el objetivo es diseñar un eficiente sistema de reconocimiento de voz humano capaz de identificar y verificar el habla humana con mayor precisión.

  Finalmente, se presenta la implementación de la técnica de identificación del hablante dependiente de texto utilizando las Máquinas de Vectores de Soporte de Dominio y MFCC, usando la Técnica de Aprendizaje de Optimización Mínima Secuencial (SMO) para SVM que mejora la actuación sobre técnicas tradicionales, con una muestra de 20 veces la palabra "cero" por cada locutor y un total de 8 locutores, cuyo resultado tuvo una tasa de éxito del 91.88% al utilizar Chunking SVM y 95% al utilizar SMO SVM.

- **David Bonomo Laynez, "Sistemas de verificación automática de locutor", Universidad de Sevilla, España, 2012.** Se trata de una revisión bibliográfica sobre las diferentes técnicas de Verificación Automatica de Locutor que se estudiaron y analizaron en la última decada, en concreto, se aborda el GMM (Modelo de Mezclas Gaussianas), SVM (Support Vector Machine), NAP (Nuissance Attribute Projection) y FA (Factor Analysis), donde el objetivo es comparar la eficiencia de las cuatro técnicas ante una misma base de

datos, lo que proporciona una base sólida para investigar futuras técnicas en la Verificación Automática de Locutor.

Finalmente, después de un análisis detallado sobre los cuatro sistemas, se llevó a cabo la parte experimental siguiendo el protocolo de evaluaciones del NIST (National Institute of Standars and Technology), donde los resultados captaron los avances y mejoras significativas de los sistemas de Verificación Automática de Locutor en los últimos años.

- **Xinyu Zhou, Yuxin Wu and Tiezheng Li, "Digital signal processing: Speaker recognition final report", Tsinghua University, China, 2014.** Se trata de una revisión bibliográfica sobre el estudio del reconocimiento de locutor (speaker recognition) usando los Modelos de Mezclas Gaussianas (GMM), que consiste en el reconocimiento de un individuo a través de su voz continua (múltiples locutores emitiendo sonido de voz en serie), quiere decir que el sistema es capaz de reconocer (identificar y verificar) al emisor (locutor o speaker) de la señal de voz por medio de un sistema desarrollado en Python, c++ y Matlab, donde no se encuentra el objetivo pero se puede discernir que es el desarrollo de un sistema de reconocimiento automático de locutor para individuos con habla continuo y uso de corpus chino.

  Finalmente, se presenta la implementación mejorada del Modelo de Mezclas Gauseanas GMM, al usar un método de clustering K-meansII, filtro de bancos, CRBM y JFA, obteniendo resultados optimos a medida que incrementa el numero de locutores, así con 512 000 características su método es 5 veces mejor frente al GMM desarrollado por Scikit-learn, este GMM desarrollado en este paper muestra con 20s de sonidos de voz en el registro training y 50 muestras para cada usuario, obteniendo una superior aceptación al mostrado por el GMM de Scikit –learn.

3.1.2 Antecedentes nacionales

- **Jhon Dennis Auccapuma Gamarra, Errol Wilderd Mamani Condori, "Identificación de locutor usando codebooks de coeficientes cepstrales en las frecuencias de mel y modelos ocultos de markov", Universidad San Antonio Abad del Cusco, 2016.** Se trata de una revisión bibliográfica sobre la identificación de locutor mediante palabras aisladas usando los Modelos Ocultos de Markov(HMM), Coeficientes Cepstrales en Frecuencias Mel y Vector de Cuantificación, donde el objetivo es implementar los algoritmos para identificar a un locutor usando codebooks de Coeficientes Cepstrales en las Frecuencias Mel con los Modelos Ocultos de Markov y reestimar los parámetros de los módulos pre procesamiento, extracción de características, post procesamiento y la cuantificación vectorial para obtener buenos resultados en la identificación.

  Finalmente, se presenta la implementación de un identificador de locutor usando Coeficientes Cepstrales en Frecuencias Mel, Vector de Cuantificación y los Modelos Ocultos de Markov (HMM). La muestra es de 20 personas (12 varones y 8 mujeres), con 10 palabras por persona, cada palabra con una duración de 2.5 segundos. Los resultados obtenidos en un ambiento sin ruido de fondo tiene una tasa de rendimiento de 90% para un grupo reducido y cerrado de 4 locutores, además la tasa de identificación tiende a disminuir a medida que se incrementan locutores, también cuando las frases y palabras contienen ruido blanco al momento de la prueba y modelamiento.

3.2 Marco teórico

3.2.1 El sonido

Al sonido audible se le puede definir como la decodificación que realiza el cerebro de las vibraciones que percibe a través del oído, el cual está formado por las ondas que fluyen en el aire y que el oído convierte en ondas mecánicas, luego en impulsos nerviosos que el cerebro pueda percibir y procesar.

Por definición, el sonido es una onda mecánica longitudinal que se propaga a través de un medio elástico (gaseoso, líquido o solido) como una onda de presión por el movimiento de átomos o moléculas (E Tippens, y otros, 2007).

El sonido se puede representar como una suma de curvas sinusoides con un factor de amplitud diferente, con las siguientes caracteristicas, apreciadas en la figura 1:

- **λ** es longitud de onda;
- **f** es frecuencia;
- **T** es periodo;
- **A** es amplitud.

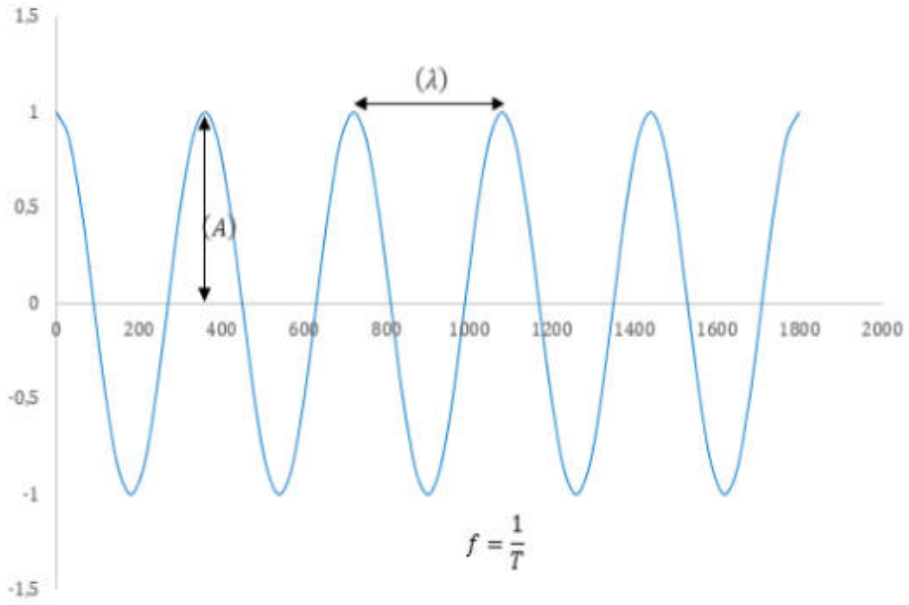

Figura 1 — Representación del sonido
Extraído de Rueda Rojo, 2011

3.2.2 La voz

La voz es el sonido que produce el ser humano cuando se expulsa el aire a través de la laringe haciendo que vibren las cuerdas vocales. Se consideran órganos de producción de la voz: la faringe laríngea, la faringe oral, la cavidad oral, la faringe nasal y la cavidad nasal (Garretón Vender, 2007).
La diferencia de voces en las personas se encuentra en la morfología del tracto vocal, características de las cuerdas vocales, la velocidad del habla,

los efectos prosódicos y el dialecto. La morfología del tracto vocal puede ser estimada de la forma del espectro de señal de voz, es posible distinguir las características de la localización de las frecuencias formantes o la oscilación espectral. (P. Campbell, 1997)

3.2.3 Arquitectura de la identificación de locutor

La arquitectura de la identificación de locutor está conformada por dos fases bien diferenciadas: el entrenamiento y el test. Estos comparten una estructura similar en cuanto a los módulos que las componen pero tienen una función diferente.

a) La fase de entrenamiento

Consiste en registrar fragmentos de voz de locutores mediante un micrófono para extraer sus características, generar su modelo correspondiente y guardarlo en una base de datos para ser utilizado en la fase de test.

b) La fase de test

Consiste en registrar el extracto de voz del locutor a identificar y extraer las características para generar su modelo y así poder compararlas con los modelos anteriormente almacenados en la base de datos. Finalmente, después de una evaluación se determinará cuál modelo obtuvo posibles coincidencias con el modelo del test, obteniendo el locutor buscado.

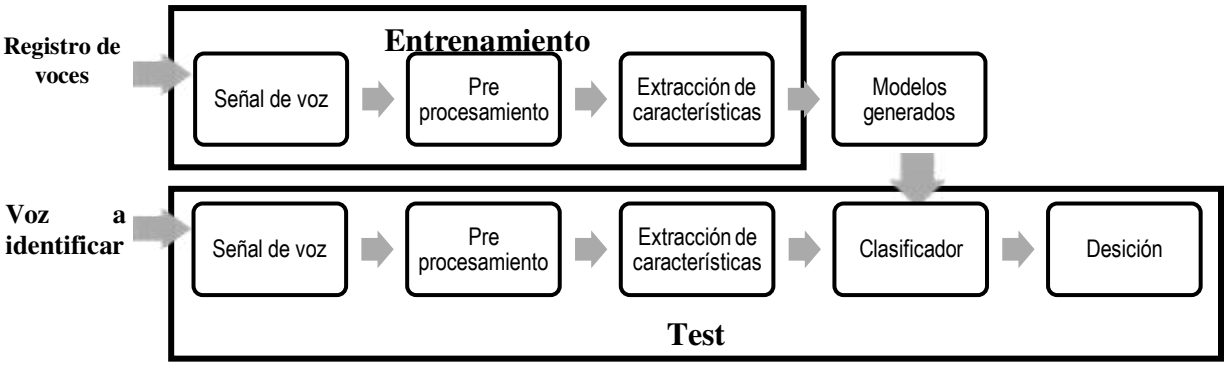

Figura 2 — Diagrama de proceso para la arquitectura de la identificación de locutor, se divide en 2 fases: Entrenamiento y Test. Elaborado en base a la ejecución de esta investigación

Para desarrollar la identificación de locutor, se llevaron a cabo los procesos mostrados en la Figura 3. En este diagrama se incluyen las fases correspondientes al entrenamiento y test, como son la adquisición de la señal de voz a través de un computador, el pre procesamiento de las mismas, la extracción de características, como los coeficientes cepstrales de las Frecuencias de Mel (MFCC) que son utilizados para el entrenamiento y por lo tanto en la tarea de clasificación. Finalmente se verifica la respuesta de identificación de locutor obteniéndola en términos de probabilidad, que indica el grado de aceptación frente a la señal de voz presentada.

Analizaremos todos los pasos necesarios generalmente usados antes de aplicar los algoritmos de modelamiento. (Bonomo Laynez, 2012)

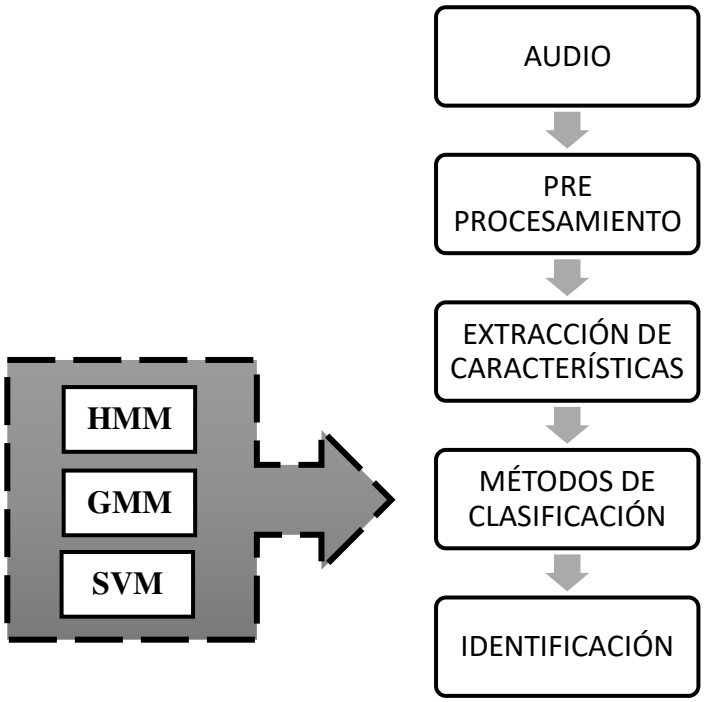

Figura 3 — Esquema de la identificación de voz, elaborado en base a la ejecución de esta investigación

3.2.3.1 Captura de voz

En el procesamiento del habla se convierten las señales analógicas en una señal digital x[n], donde n es un índice en el tiempo. Las señales digitales se aproximan a las señales analógicas en mayor o menor medida según la tasa de muestreo y la profundidad en bits.

a) Tasa de Muestreo (Sample rate)

La tasa de muestreo es el número de muestras que se toman en intervalos de tiempos regulares para convertir la señal analógica en señal digital. Se le denomina frecuencia de muestreo a la cantidad de muestras tomadas de una onda. A una cantidad mayor de frecuencia de muestreo se consigue un sonido más parecido al original.

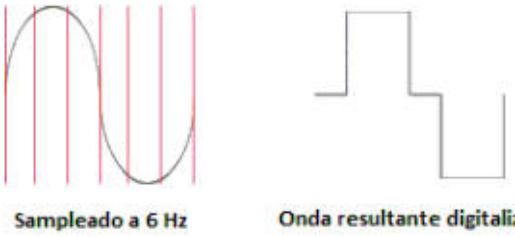

Figura 4 — Ejemplo de frecuencia de muestreo 6Hz
Extraído de Multison Online, 2016

Cuanto más alta sea la tasa de muestreo, la captura del sonido resultará más precisa y, en consecuencia, el sonido digital será de mayor calidad.

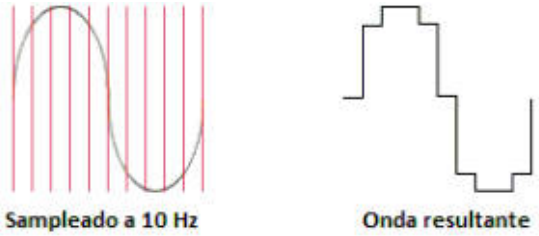

Figura 5 — Ejemplo de frecuencia de muestreo 6Hz
Extraído de Multison Online, 2016

b) Profundidad de bit o Bit resolution

La profundidad de bits es el rango dinámico de sonido, que divide el eje horizontal de la forma de onda (la amplitud). Es el número de bits utilizados para almacenar cada muestra de la señal analógica. Si la tasa de bits es mayor, se disponen de más posiciones para ubicar la muestra tomada del instante de la onda.

Una resolución de 8-bits proporciona 256 (2^8) niveles de amplitud, una resolución de 16-bits alcanza 65536 (2^{16}), mientras que una resolución de 24-bits proporciona

16777216 (2^{24}). Un audio digital tendrá mejor calidad cuanto mayor sea su resolución.

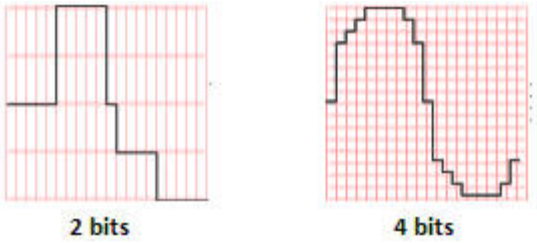

Figura 6 — Ejemplo de resolución de 2 bits vs resolución de 4 bits

Extraído de Multison Online, 2016

c) PCM (Pulse Code Modulation)

La Modulación por Impulsos Codificados es una representacion digital de una señal analógica, un procedimiento de modulación utilizado para transformar una señal analogica en una secuencia de bits (señal digital). (Villalobos Vives , y otros, 2010)

Los flujos PCM tienen dos propiedades básicas que determinan su fidelidad a la señal analógica original: la frecuencia de muestreo, es decir, el número de veces por segundo que se tomen las muestras; y la profundidad de bit, que determina el número de posibles valores digitales que puede tomar cada muestra.

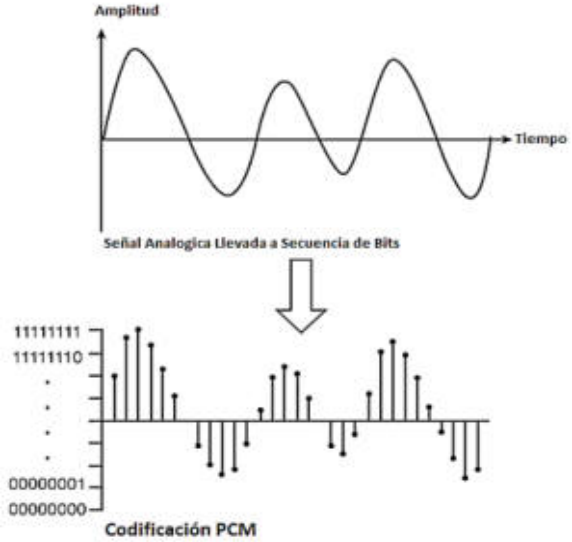

Figura 7— PCM (Modulación por impulsos codificados)

Extraído de Hub electronics, 2015

3.2.3.2 Pre-procesamiento

a) Detección de punto final y eliminación de silencio

Una vez capturada el audio el siguiente paso en el tratamiento de la señal es la eliminación de silencio como al inicio y al final de la señal, entre las palabras de una frase o al final de una señal. Si el silencio está presente en la trama, los recursos de modelamiento se gastan en partes que no son importantes para la identificación, entonces, el silencio presente debe eliminarse antes de su posterior procesamiento.

Hay varias maneras de hacer esto; las más populares son el **Short Time Energy** (Energías de intervalo de tiempo corto) y **Zeros Crossing Rate** (Tasa de cruce por ceros). Pero estos tienen su propia limitación en cuanto a establecer umbrales como una base ad hoc. El algoritmo que utilizó (Goutam, y otros, 2005) establece propiedades estadísticas del ruido de fondo, así como los aspectos fisiológicos de la producción del habla y no asume ningún umbral ad hoc. Asume que el ruido de fondo presente en las declaraciones es de naturaleza gaussiana (ruido blanco).

- **Zero crossing rate.-** Es un algoritmo utilizado para detección de punto final e inicial de un tramo de voz sonora y el razonamiento generalizado es que cuando la tasa de cruces por cero es baja, existe voz sonora y viceversa, cuando la tasa de cruces por cero es alta, existe voz sorda. Teniendo en cuenta que la aseveración anterior no es correcta del todo, es necesario hacer otros tipos de consideraciones entonces para el desarrollo se utiliza una fórmula de cruces por el centro denotando con positivo por encima y negativo por debajo siendo la fórmula promedio utilizada la siguiente (Hansen, y otros, 1987).

$$Z_n = \frac{1}{N}\sum_{n=m-N+1}^{m} \frac{sgn[s(n)]-sgn[s(n-1)]}{2} w(m-n)$$

Donde:

$$sgn[x(n)] = \begin{cases} -1, \ para\ x(n) < 0 \\ 0, \ para\ x(n) = 0 \\ 1, \ para\ x(n) > 0 \end{cases}$$

Además el proceso seguido gráficamente es la siguiente:

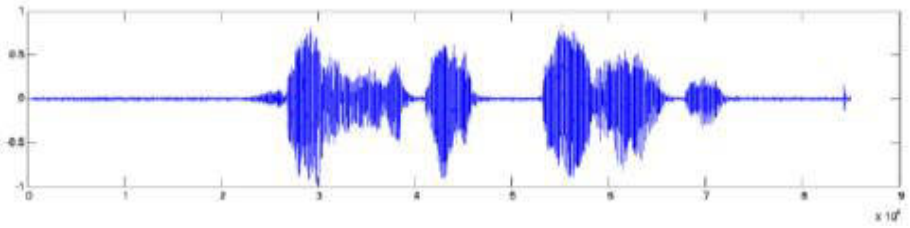

Figura 8 — Señal de entrada al sistema de Detección de Punto Final

Extraído de Aya Gamal Osman Fatma A. Mohammed Ahmed Moawad Ahmed Helmy, 2012

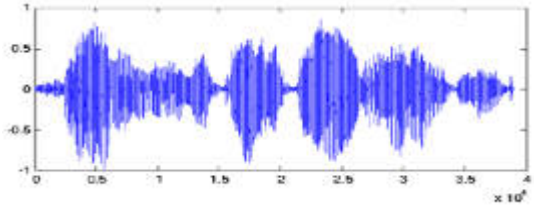

Figura 9 — Salida de la señal del sistema de Detección de Punto Final

Extraído de Aya Gamal Osman Fatma A. Mohammed Ahmed Moawad Ahmed Helmy, 2012

b) Normalización

La normalización de audio consiste en igualar al máximo el volumen promedio (o la amplitud de la señal) de una misma o varias pistas de audio, para que éstas suenen al mismo volumen general y no se escuchen unos segmentos más altos que otros (Hernández García, 2011).

- **Normalización de pico**

  Es un proceso automatizado que cambia el nivel de cada muestra para llevar el valor PCM más alto, el pico más alto o la muestra mas fuerte de una señal analógica a un nivel específico.

  Por buscar el nivel más alto, no considera el volumen aparente del contenido. En consecuencia, es utilizado en la etapa de masterización de una grabación.

- **Normalización de la sonoridad**

  La normalización de sonoridad es utilizada para llevar la amplitud media a un valor objetivo cambiando el nivel de cada muestra. Así, la sonoridad de la fuente de sonido se viene a ajustar al nivel objetivo de la sonoridad dentro de un rango dinámico donde el pico no se recorta.

3.2.3.3 Extracción de características

La extracción de características es un paso muy importante en la construccion de un identificador de voz, porque despues de haber convertido la señal de voz a una señal digital, debemos de convertirla en un vector de características.

a) MFCC (Mel Frequency Cepstral Coefficients)

Los coeficientes MFCC se introdujeron originalmente para el reconocimiento de habla y fueron posteriormente adaptados para sistemas automáticos de reconocimiento de locutor de nivel espectral. Actualmente es la parametrización más extendida para dichos sistemas. Este método utiliza un banco de filtros mel que pretende imitar el comportamiento del oído humano dando mayor importancia a las frecuencias bajas frente a las altas (Rubén Zazo Candil et al. Análisis de compensación de variabilidad en reconocimiento de locutor

aplicado a duraciones cortas), Su obtención requiere una serie de etapas que describiremos a continuación en la Figura 10.

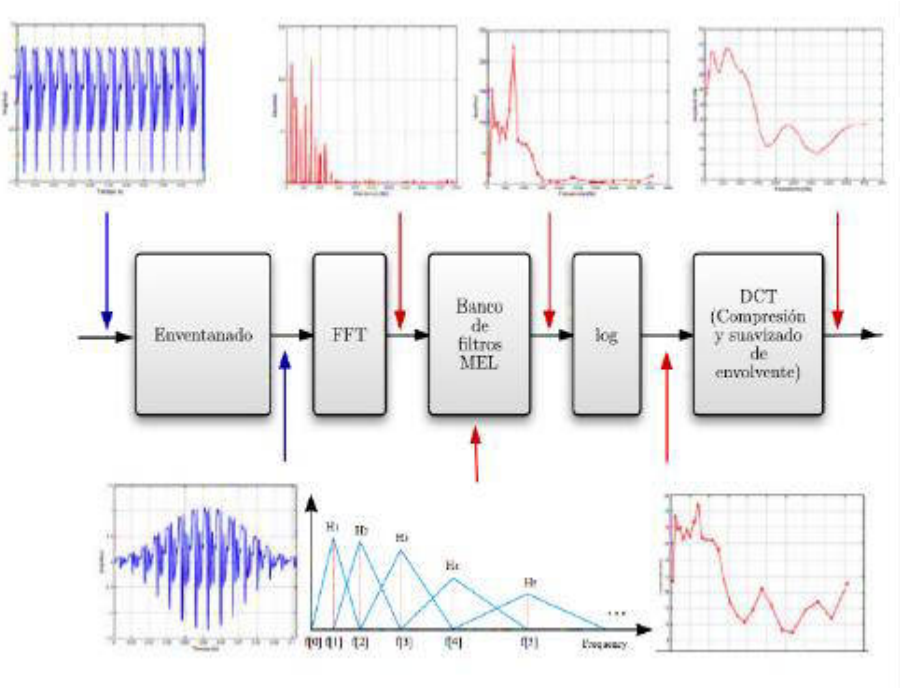

Figura 10 — Esquema de extracción de los coeficientes MFCC
Extraído de Zaso Candil, 2014

b) Entramado y enventanado

La señal de voz como un proceso aleatorio y no estacionario, muestra inconvenientes a la hora de analizarla. Sin embargo, se puede salvar teniendo en cuenta que la señal es cuasi-estacionaria a corto plazo de tiempo. Esto da lugar a un estudio denominado análisis localizado donde se obtienen segmentos o tramas de la señal de pocos milisegundos (Bonomo Laynez, 2012).

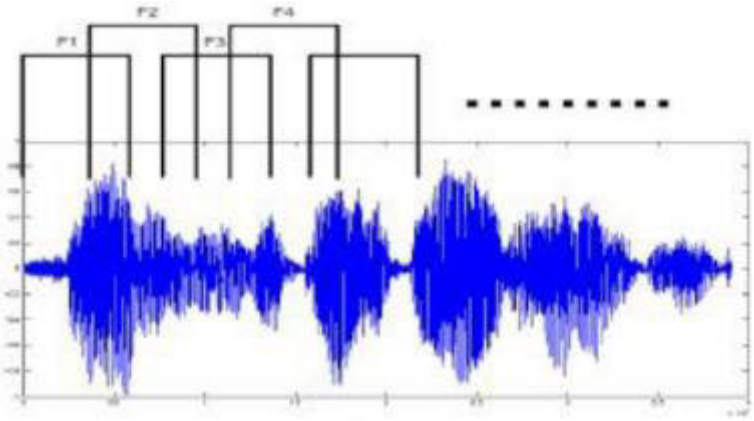

Figura 11 — Proceso de entramado de la señal
Extraído de Aya Gamal Osman Fatma A. Mohammed Ahmed Moawad Ahmed Helmy, 2012

Se le denomina enventanado al proceso donde se generan segmentos o tramas consecutivas de señal. Normalmente se trabaja con ventanas de tipo Hanning y de Hamming (algoritmo), para obtener el valor de la ventana promedio lo multiplicamos por un valor encuadrando la señal en una ventana, ver Figura 12.

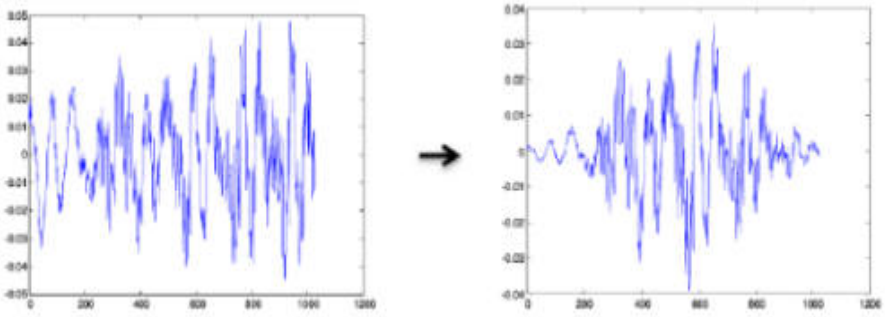

Figura 12 — Trama simple antes y después del enventanado
Extraído de Aya Gamal Osman Fatma A. Mohammed Ahmed Moawad Ahmed Helmy, 2012

c) Transformada Discreta de Fourier

De los pasos anteriores tenemos ya la información tratada para ser procesada, pero debido al análisis lógico en el dominio del tiempo (eje x) y la forma en la que se presenta no es posible

tener más datos para analizar entonces los estudios mostraron que se debe analizar en otro dominio y esta es la frecuencia de tal manera que es más fácil y con muchos datos para ser analizados, es en este punto donde se recurre a un método de transformación de dominio, del tiempo al dominio de la frecuencia y viceversa y el método más usado y eficiente es la Transformada de Fourier cuya definición dice, en esencia, que se descompone o expande una señal o función en senos y cosenos de diferentes frecuencias cuya suma corresponde a la señal original, es decir, es capaz de distinguir las diferentes componentes de frecuencia de la señal, y sus respectivas amplitudes como lo vemos en la Figura 13.

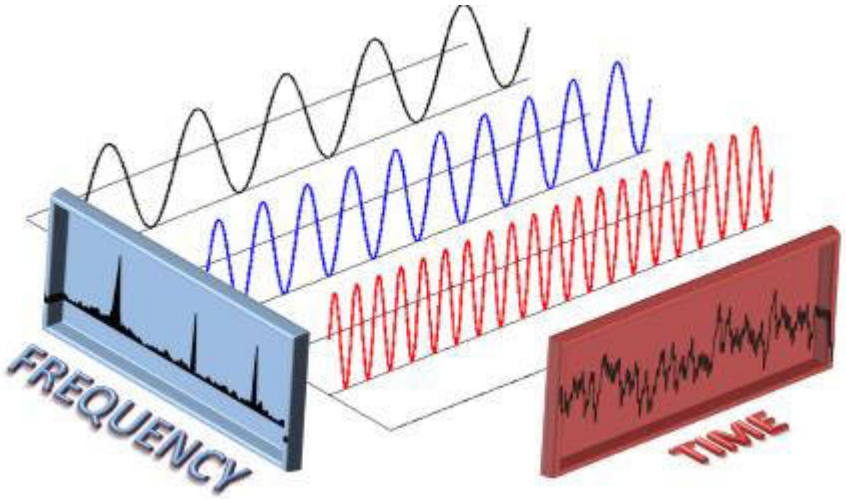

Figura 13 — Figura de la Transformada de Fourier de una señal, así como el dominio del tiempo (rojo) y dominio de la frecuencia (Azul)

Extraído de Networks at MIT group NETMIT, 2014

El algoritmo desarrollado para la Transformada de Fourier y el más óptimo encontrado es la FFT (Fast Fourier Transform) el término ingles define la Transformada Rápida de Fourier como una alternativa óptima para el desarrollo de la transformación de dominio, se tiene en cuenta que:

- Las limitaciones de la FFT emergen de las que tiene la DFT.

- La FFT es un algoritmo a iguales intervalos de espaciamiento.
- Debido a los números de operaciones menores que utiliza para ser resuelta la FFT, se logra una eficiencia.

d) Banco de filtros en escala Mel

El oído humano actúa esencialmente como un banco de filtros superpuestos paso banda. La percepción humana está basada en la Escala Mel. De este modo, la mejor aproximación para simular la percepción humana es construir un banco de filtros triangulares con un ancho de banda dado por la Escala Mel y pasar las magnitudes de los espectros, a través de estos filtros y así obtener el espectro de frecuencias Mel (Sayed Jaafer, y otros, 2012)

Pues bien, nosotros utilizamos un banco de filtros triangular con M filtros (m =1, 2,…, M) y N puntos de la Transformada Discreta de Fourier (DFT) (k = 1, 2, … , N), donde, Hm[k], es la magnitud de frecuencia correspondiente de los filtros datos por:

$$H_m[k] = \begin{cases} 0, & k < f[m-1] \\ \frac{(k-f[m-1])}{(f[m]-f[m-1])}, & f[m-1] \le k \le f[m] \\ \frac{(f[m-1)-k]}{(f[m+1]-f[m])}, & f[m] \le k \le f[m+1] \\ 0, & k > f[m+1] \end{cases}$$

Donde:

k *es el dato del* $k-$*esímo elemento de la muestra por trama;*
m *es el numero de filtros;*
$f[m]$ *es la frecuencia uniformemente espaciada en la escala mel;*
$H_m[k]$ *es el dato del* $k-$*esímo elemento del banco.*

La ecuación anterior fue desarrollada de acuerdo al Journal (World of Computer Science and Information Technology Journal ISSN, 2012)

e) Cepstral Mean Subtraction(CMS)

Una de las razones primordiales para degradar o empeorar los resultados y el rendimiento de un sistema de reconocimiento de locutor se debe a la variabilidad acústica entre las locuciones de entrenamiento y las de test. Esta variabilidad, al margen de la variabilidad del locutor de una grabación a otra, se debe principalmente a la variabilidad del canal, originada por distorsiones en el micrófono, teléfono, medio de transmisión, o en las propias condiciones ambientales en el momento de la grabación. La utilización de técnicas de compensación de canal, o bien sobre el audio, o bien sobre el modelo de características mejora las técnicas y los resultados del reconocimiento. Existen diferentes técnicas orientadas a la disminución o eliminación de la variabilidad del canal (Gonzales Domínguez, 1998).

De acuerdo a esto el efecto de canal es eliminado por restando los coeficientes Mel-cepstrum con:

$$mc_i(q) = C_i(q) - \frac{1}{M}\sum_{i=1}^{M} C_i(q), q \qquad q = 1,2, \dots ,12$$

Donde:

M es el número del tramas;

i es el i-esímo coeficiente de la i-esíma trama;

$C_i(q)$ es el q-esímo coeficiente característico MFCC;

$mc_i(q)$ es el q-esímo coeficiente Normalizado MFCC.

f) Cálculo de los parámetros delta

Una vez obtenido los coeficientes MFCC y tomando en cuenta los efectos del canal, también debemos obtener otras características importantes en la extracción como son otras características a las que denominaremos Deltas, siguiendo esto las ecuaciones son descritas de la siguiente forma.

En la coarticulación de los fonemas, se integran más coeficientes, que son los MFCC-Delta (ΔMFCC) y los MFCC-Delta-Delta (o ΔΔMFCC) (Furui, 1981), muy aparte de los MFCC que permiten considerar la variabilidad de un interlocutor en el momento de dialogar y representan la evolución temporal de los fonemas en su transición a otros fonemas. A los ΔMFCC se les denomina Coeficientes de Rapidez, ya que miden la alteración de los coeficientes MFCC sobre un momento de tiempo. De igual manera, a los que representan la variación de los Delta MFCC en un instante de tiempo, se les conoce como Coeficientes de Aceleración ΔΔMFCC.

Si Ci[m] representa el componente m del vector MFCC asociado a la trama o ventana i-ésima, se calculará el valor de los coeficientes dinámicos con la siguiente expresión:

$$d(t) = \Delta f_k[i] = f_{k+M}[i] - f_{f-M}[i]$$

Donde:

i es el tiempo del particular valor cepstral;

$\Delta f_k[i]$ es la diferencia delta k-esíma de la señal en un tiempo i;

M es el valor independiente del tamaño de la ventana.

El método de diferencia es simple, pero desde que actúa como un filtro de paso alto en el dominio del parámetro, tiende a amplificar el ruido. La solución para esto es la regresión lineal, por ejemplo: El polinomio de primer orden, la solución de mínimos cuadrados es fácilmente mostrada dada la siguiente forma:

$$\Delta f_k[i] = \frac{\sum_{m=-M}^{M} m f_{k+M[i]}}{\sum_{m=-M}^{M} m^2}$$

3.2.4 Los Modelos de Mezclas Gaussianas (GMM)

3.2.4.1 Definición de un GMM

Según Alberto García Herrero, define a GMM como, Sea Y= $[Y_1, Y_2, \cdots, Y_D]t$ una variable aleatoria real D-dimensional. Se dice que la distribución de Y sigue una distribución mezcla finita si su función densidad de probabilidad (fdp) se puede escribir como una combinación lineal de fdp's elementales.

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{i=1}^{I} \alpha_i p(\boldsymbol{y}|C = i, \boldsymbol{\beta}_i), \qquad i \in \{1, \dots, I\}$$

Donde I representa el número de distribuciones elementales (componentes) de la mezcla, C= $1,2,\dots,I$ y θ representa el conjunto de parámetros

$$\theta = \{\alpha_1, \dots, \alpha_I, \beta_1, \dots, \beta_I\}$$

Siendo $\beta = \{\beta_1, \dots, \beta_I\}$ el conjunto de parámetros asociados a cada distribución de la mezcla y $\alpha = \{\alpha_1, \dots, \alpha_I\}$ la probabilidad o peso de cada distribución de la mezcla.

Cuando las distribuciones que componen la mezcla son gaussianas, la función densidad de probabilidad (primera ecuación en esta definición) se conoce como mezcla gaussiana. Por tanto una mezcla gaussiana es una distribución probabilística cuya fdp (función de densidad de probabilidad) es una combinación lineal de distribuciones gaussianas.

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{i=1}^{I} \alpha_i N(\boldsymbol{y}|C = i, \boldsymbol{\beta}_i), \qquad i \in \{1, \dots, I\}$$

Siendo:

$$N(\boldsymbol{y}|C = i, \boldsymbol{\beta}_i) = \frac{1}{(2\pi)^{\frac{D}{2}} |\sum_i|^{\frac{1}{2}}} exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu}_i)^T \sum_i^{-1} (\boldsymbol{y} - \boldsymbol{\mu}_i)\right)$$

Y los parámetros de cada gaussiana:

$$\beta_i = \{\mu_i, \sum_i\}$$

Donde $\boldsymbol{\mu_i} \in \mathbb{R}^D$ y $\sum_i \in \mathbb{R}^{DxD}$ son la media y la matriz de covarianza de la componente i-ésima.

3.2.4.2 Restricciones de los parámetros de una mezcla de gaussianas

Los parámetros α y $\sum_i$ presentan una serie de restricciones a tener en cuenta. Las probabilidades de la mezcla, deben verificar (García Herrero, 2015):

$$\alpha_i \geq 0, i \in \{1, \dots, I\}, \qquad \sum_{i=1}^{I} \alpha_i = 1$$

Y las matrices de covarianza deben cumplir las siguientes restricciones:

$$\sum_i = \sum_i^T$$

$$\sigma_{jj}^2 \geq 0, \qquad j \in \{1, \dots, D\}$$

Las matrices de covarianza deben ser simétricas, además en los elementos de la diagonal se encuentran las varianzas y deben ser no negativas.

3.2.4.3 Estimación de máxima verosimilitud

Sea un conjunto de datos $\mathcal{Y} = \{y_1, y_2, \cdots, y_j\}$, donde $y_j \in \mathbb{R}^D$ es una de las J realizaciones independientes e idénticamente distribuidas (i.i.d.) de la variable aleatoria **Y**. Dado un modelo estadístico, la estima ML proporciona las estimaciones de los parámetros del modelo a partir de un conjunto de datos.

La verosimilitud de $\mathcal{Y}$ es:

$$L(\boldsymbol{\theta}) = \prod_{j=1}^{J} p(\boldsymbol{y}_j|\boldsymbol{\theta})$$

La función verosimilitud es una función de los parámetros $\boldsymbol{\theta}$ del modelo. El estimador ML para $\boldsymbol{\theta}$.

$$\widehat{\boldsymbol{\theta}}_{ML}(\boldsymbol{Y}) = argmax_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = argmax_{\boldsymbol{\theta}} \prod_{j=1}^{J} p(\boldsymbol{y}_j|\boldsymbol{\theta})$$

Habitualmente se maximiza el logaritmo de la verosimilitud por ser más sencillo de resolver analíticamente. Esto es posible debido a que el logaritmo es una función monótona creciente:

$$\widehat{\boldsymbol{\theta}}_{ML}(\boldsymbol{Y}) = argmax_{\boldsymbol{\theta}} \sum_{j=1}^{J} \log\, p(\boldsymbol{y}_j|\boldsymbol{\theta})$$

Aplicado al modelo de mezclas gaussianas:

$$\widehat{\boldsymbol{\theta}}_{ML}(\boldsymbol{Y}) = argmax_{\boldsymbol{\theta}} \sum_{j=1}^{J} \log\{\sum_{i=1}^{I} \alpha_i N(\boldsymbol{y}_j|C = i, \boldsymbol{\beta}_i)\}$$

Se trata de un problema de optimización difícil, no existe una forma cerrada para $\hat{\theta}_{ML}$.

3.2.4.4 En el reconocimiento de voz

Los modelos de mezclas gaussianas han sobresalido en el desarrollo de los sistemas de reconocimiento de voz independiente de texto. Estos resultan adecuados debido a que no modelan el texto que se dice, sino que se basa en las características espectrales que tiene la voz para diferenciar a los locutores. Su utilización en el reconocimiento de locutor independiente de texto fue publicado por Reynolds y Rose desde 1990. (Esteve Elizalde, 2007).

En el reconocimiento de locutor se busca que dado un segmento de entrenamiento que presentamos mediante $\boldsymbol{X} = \{\vec{x}_t\}_{t=1,\dots,N_X}$ y uno de evaluación que identificamos como $\boldsymbol{Y} = \{\vec{y}_t\}_{t=1,\dots,N_Y}$, en cualquier sistema de identificación existen dos posibles hipótesis (Bonomo Laynez, 2012):

- H_0: los segmentos **X** e **Y** pertenecen a la misma persona.
- H_1: los segmentos **X** e **Y** no pertenecen a la misma persona.

Si X representa al modelo predicho en la etapa de entrenamiento con el segmento **X**, durante la etapa de evaluación, dado **Y**, quisiéramos saber cuál de las dos hipótesis es más probable. Es decir, a partir de un criterio estadístico, queremos comparar $p(X|\boldsymbol{Y})$ y $p(\bar{X}|\boldsymbol{Y})$ (donde $\bar{X}$ se conoce como el modelo alternativo a X). Por consiguiente, se aceptará la hipótesis si:

$$p(X|\boldsymbol{Y}) > p(\bar{X}|\boldsymbol{Y})$$

Por medio del teorema de Bayes equivale afirmar que:

$$\frac{p(\boldsymbol{Y}|X)\,p(X)}{p(\boldsymbol{Y})} > \frac{p(\boldsymbol{Y}|\bar{X})p(\bar{X})}{p(\boldsymbol{Y})},$$

$$p(\boldsymbol{Y}|X)\,p(X) > p(\boldsymbol{Y}|\bar{X})p(\bar{X}),$$

$$\frac{p(\boldsymbol{Y}|X)}{p(\boldsymbol{Y}|\bar{X})} > \frac{p(\bar{X})}{p(X)},$$

$$\frac{p(\boldsymbol{Y}|X)}{p(\boldsymbol{Y}|\bar{X})} > \theta$$

$p(X)$ Denota la probabilidad a priori de tener un juicio en el cual los datos de entrenamiento y evaluación pertenecen al mismo locutor ("*true trial*"), de tal manera que el cociente entre $p(X)$ y $p(\bar{X})$ puede considerarse como el umbral θ de nuestro sistema. Sin embargo, depende a la aplicación en la que se trabaje, será de interés un umbral más alto o más bajo, con lo que ese cociente es un parámetro ajustable a los intereses de la aplicación. Es de interés calcular dos modelos estadísticos, X y $\bar{X}$ que denoten al usuario y al modelo alternativo, respectivamente, de tal forma que se pueda evaluar ambos modelos y obtener un resultado que facilite escoger una de las dos posibles hipótesis.

Dado un modelo estadístico λ de un locutor o un conjunto de locutores (se puede representar con X o $\bar{X}$), la probabilidad de que un vector cepstral de evaluación $\vec{y}_t$ pertenezca a dicho modelo se representa por medio de una combinación lineal de distribuciones de probabilidad gaussianas (la dimensión del vector de

características $\vec{y}_t$) (D.A. Reynolds, 1992), se muestra en lo siguiente:

$$p(\vec{y}_t|\lambda) = \sum_{i=1}^{M} w_i p_i(\vec{y}_t)$$

Que representa la cantidad de componentes gaussianas, sus pesos sujetos a la restricción $\sum_{i=1}^{M} w_i = 1$ y $p_i(\vec{y}_t)$ que se expresa de la siguiente forma:

$$p_i(\vec{y}_t) = \frac{1}{(2\pi)^{D/2}|\sum_i|^{1/2}} e^{-\left(\frac{1}{2}\right)(\vec{y}_t - \vec{\mu}_i)' \sum_i^{-1} (\vec{y}_t - \vec{\mu}_i)} \qquad i = 1, \dots, M.$$

Sin olvidar que se tiene una restricción adicional de:

$$\int_{-\infty}^{\infty} p_i(\vec{y}) d\vec{y} = 1$$

Entonces, para cada componente gaussiana se tendrá su correspondiente vector de medias y matriz de covarianza, es decir para el modelo de locutor o el modelo alternativo, de esta manera al modelo se le identificará mediante la fórrmula $\lambda = \{\omega_i, \vec{\mu}_i, \sum_i\}$ $i = 1, \dots, M.$

a) **Ejemplo sobre la representación de modelos con componentes gaussianas**

El análisis esta centrado en una variable mono-dimensional, en el lado izquierdo está el histograma que corresponde al primer componente de los vectores de características-dimensionales de un fichero de entrenamiento de un locutor. Podemos comprobar cómo con 4 o 32 componentes gaussianas y con sus respectivos pesos generamos la distribución estadística del primer componente cepstral del locutor.

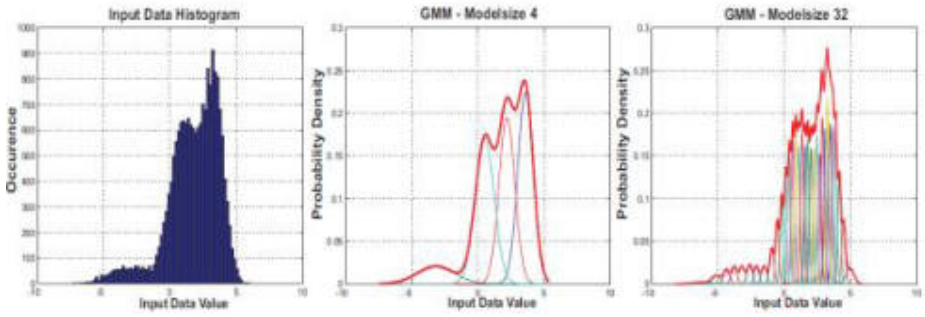

Figura 14 — Modelo mono-dimensional de mezcla gaussiana con una distribución de entrada (histograma) y dos aproximaciones GMM de tamaños 4 y 32

Extraído de Harald Priewald, 2009

A continuación, se trabaja con una variable aleatoria bidimensional gaussiana, ilustrando el mismo ejemplo con los dos primeros componentes de los vectores de características:

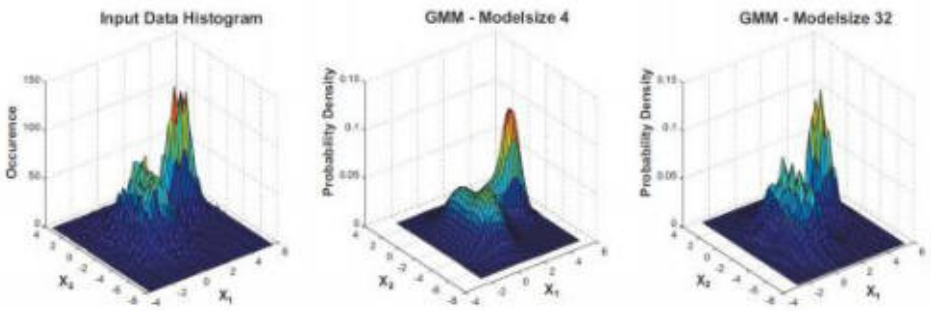

Figura 15 — Modelo bidimensional de mezcla gaussiana con una distribución de entrada (histograma) y dos aproximaciones GMM de tamaños 4 y 32

Extraído de Harald Priewald, 2009

Entonces, dado un modelo de mezclas gaussianas se puede calcular la probabilidad de que un vector de test corresponda a dicho modelo.

3.2.4.5 GMM-UBM

Reynolds en 1995 introdujo la primera técnica contemporánea en el campo de la verificación de locutor, esta vino a ser importante

porque varias aplicaciones actuales toman en cuenta varios principios de GMM-UBM.

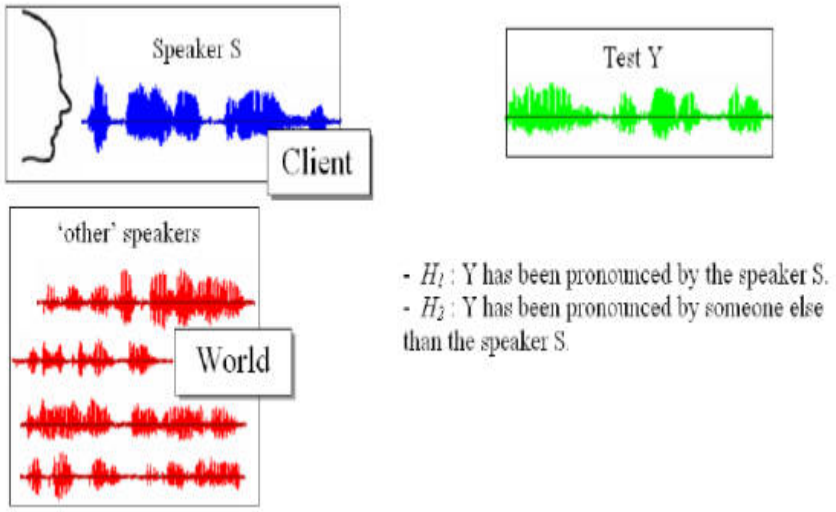

Figura 16 — Tres tipos de grabaciones necesarias en GMM-UBM

Extraído de B. Fauve, 2009

El modelo alternativo identificado sería generado por un conjunto de segmentos pertenecientes a numerosos locutores, así se generaría un modelo que represente a la población mundial (GMM -UBM). El modelo representará fielmente las características en común de los locutores que participaran en el sistema de verificación, si estos provienen de personas cuyo segmentos de audio contiene un alto grado de variación acústica. Estos audios serán de otras muchas personas con excepción de las personas que participaran en el entrenamiento y testeo, para captar las características acústicas de la población mundial. De esta manera, en el archivo de entrenamiento de un locutor, se adaptan componentes gaussianas del UBM para generar el modelo correspondiente, la Figura 4 muestra que obtenido el modelo universal (color rojo) por medio de datos pertenecientes a varios locutores, se necesita modificar parámetros de las mezclas gaussianas del UBM para conseguir el modelo GMM de una persona.

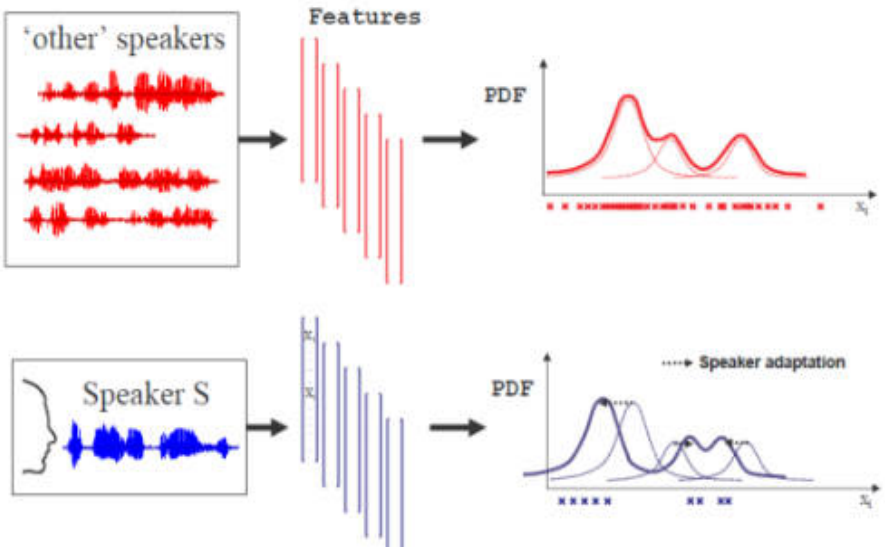

Figura 17 — Generación del modelo universal de mezcla gaussiana (en rojo), fase de entrenamiento para un locutor adaptando ciertos parámetros del UBM (en azul)
Extraído de B. Fauve, 2009

3.2.5 Los Modelos Ocultos de Markov (HMM)

3.2.5.1 Definición de un HMM.

Un modelo oculto de Markov es un autómata de estados finitos capaz de producir a su salida una secuencia de símbolos observable (Lawrence R., 1989). El autómata está formado por un conjunto de estados y evoluciona pasando de un estado a otro de forma probabilística. Los estados están conectados unos a otros por arcos de transición, con probabilidades asociadas a cada arco. Cada estado tiene asociada una función de densidad de probabilidad que define la probabilidad de emitir una observación cada vez que se produce una transición desde dicho estado del HMM. Por tanto, un HMM consta de dos procesos estocásticos: la producción de símbolos y la secuencia de los estados en la evolución del mismo. De ellos, sólo la producción de símbolos es observable. Por este motivo a este autómata se denomina Modelo Oculto de Markov. En la Figura 18 podemos ver la representación de un HMM con tres estados generando una posible secuencia de símbolos observables.

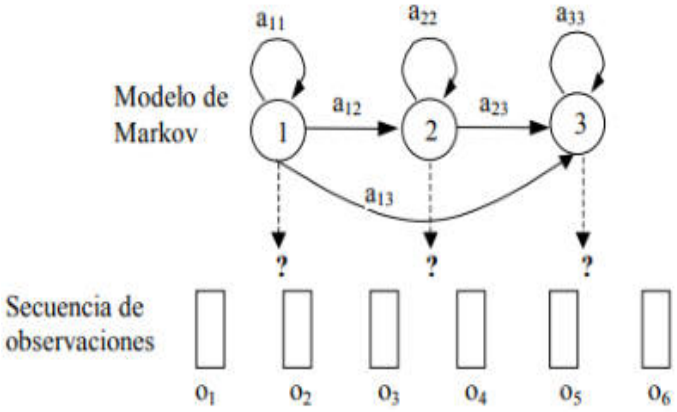

Figura 18 — Representación de un HMM con tres estados generando una posible secuencia de observaciones

3.2.5.2 Elementos de un HMM:

Está conformado por cinco (Lawrence R., 1989):

1. El conjunto *S* de "n" estados:

$$S = \{s_1, s_2, s_3, \dots, s_n\}$$

2. El conjunto *V* de posibles valores observables, donde "m" son los símbolos observables.

$$V = \{v_1, v_2, v_3, \dots, v_m\}$$

3. La matriz de probabilidades de transición de estados $A = \{a_{ij}\}$. Siendo una matriz cuadrada de dimensión "n", donde cada elemento a_{ij}, corresponde a la probabilidad de transición del estado s_i al s_j.

$$a_{ij} = P(q_t = s_j | q_{t-1} = s_i)$$

Donde q_t indica el estado del modelo en el instante t. Cada elemento a_{ij} debe cumplir:

$$0 \leq a_{ij} \leq 1 \qquad 1 \leq i, j \leq n$$

4. El conjunto de probabilidades de las observaciones $B = \{b(k)/1 \leq i \leq n, 1 \leq k \leq m\}$, cada b_j se define asi:

$$b_i(v) = P(x_t = v | q_t = s_i) \qquad 1 \leq i \leq n$$

Donde x_t representa el valor de la observación en el instante de tiempo t.

5. El conjunto de probabilidades iniciales $\pi = \{\pi_i\}$ donde $\pi_i$es la probabilidad de que el primer estado sea el estado s_i.

$$\pi_i = P(q_t = s_i) \qquad 1 \leq i \leq n$$

- **Ejemplo del clima en el HMM.-** Imaginemos observar el clima una vez al día y asignemosle uno de estos 3 estados:
  - Estado A: Lluvioso.
  - Estado B: Nublado.
  - Estado C: Soleado.

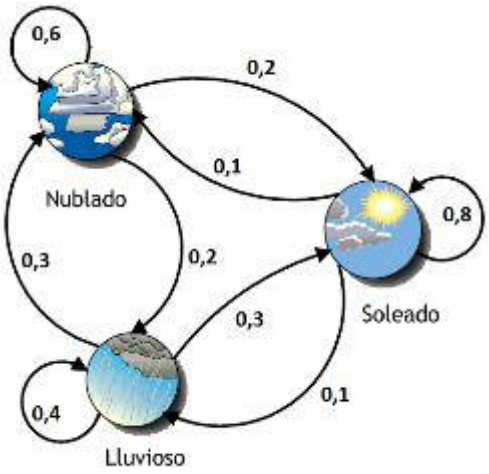

Figura 19 — Estados del clima (nublado-soleado-lluvioso)
Extraído de Aguilera Bonet

Para predecir el clima utilizaremos un Modelo Oculto de Markov simple de 3 estados, así conoceremos la probabilidad del cambio de clima entre días. Esto será proporcionado por la matriz de transición de estados:

$$A = \begin{bmatrix} 0{,}4 & 0{,}3 & 0{,}3 \\ 0{,}2 & 0{,}6 & 0{,}2 \\ 0{,}1 & 0{,}1 & 0{,}8 \end{bmatrix}.$$

También se puede definir unas probabilidades iniciales, o fijar que el día primero que se midió estaba soleado.

Así, por ejemplo, se puede calcular los probables climas de los próximos 7 días, el tiempo sería "Soleado (hoy) – Soleado – Soleado – Lluvioso – Lluvioso – Soleado – Nublado - Soleado". De esta manera, delimitamos la secuencia observada O como

O = 3, 3, 3, 1, 1, 3, 2, 3.

Y computamos la probabilidad:

$$P(O|Modelo) = p(3,3,3,1,1,3,2,3|Modelo)$$
$$= p(3).p(3|3).p(3|3).p(1|3).p(1|1).p(3|1).p(2|3).p(3|2)$$
$$= \pi_3.a_{33}.a_{33}.a_{31}.a_{11}.a_{13}.a_{32}.a_{23}$$
$$= 1.(0{,}8).(0{,}8).(0{,}1).(0{,}4).(0{,}3).(0{,}1).(0{,}2)$$
$$= 1{,}536.10^{-4}$$

3.2.5.3 Los tres problemas básicos de los HMM

Para utilizar los Modelos Ocultos de Markov en un sistema de reconocimiento, se requiere resolver tres problemas (Lawrence R., 1989):

a) **Evaluación:** Dada una secuencia de observaciones $X_1^T = x_1 x_s \dots x_T$ 1 y un modelo λ, se busca cómo evaluar la probabilidad $P(X_1^T|\lambda)$ de que la secuencia observada haya sido producida por dicho modelo.

La solución al problema de evaluación permitirá evaluar la probabilidad de generación de una secuencia de observaciones por un modelo. Esta probabilidad puede utilizarse para clasificar las secuencias de observaciones, lo que constituirá la base de cualquier sistema de reconocimiento basado en HMM. Un algoritmo eficiente para evaluar este problemas se denomina ***Adelante-Atrás (Forward-Backward).***

b) **Estimación:** Dada una secuencia de observaciones $X_1^T = x_1 x_s \dots x_T$ y un modelo λ, cómo elegir los parámetros del modelo $\lambda = (\pi, A, B)$ para que la probabilidad de generación de dicha secuencia por el modelo sea óptima.

La solución del problema de estimación permitirá extraer información sobre el proceso oculto, al obtener la secuencia óptima de estados. Esta información puede utilizarse para interpretar el significado de los estados del modelo, así como para la extracción de parámetros estadísticos sobre dichos estados, como pueden ser las duraciones medias de los mismos. Un algoritmo eficiente para evaluar este problema es el denominado ***Viterbi***.

c) **Decodificación:** Dada una secuencia de observaciones $X_1^T = x_1 x_s \dots x_T$, cómo obtener la secuencia de estados $Q_1^T = q_1 q_s \dots q_T$ que mejor explica la generación de la secuencia por parte del modelo λ.

La solución del problema de decodificación permitirá optimizar los parámetros del modelo para describir mejor cómo se produce una secuencia de observación dada. Un algoritmo eficiente para evaluar se denomina algoritmo ***Baum-Welch***, el cual garantiza la convergencia uniforme hacia un máximo local de la función probabilidad de generación.

3.2.6 Las Máquinas de Vectores Soporte (Support Vector Machines, SVMs)

3.2.6.1 Definición

Las Máquinas de Vectores soporte (SVM) son un método de clasificación que cumple tareas de clasificación mediante la construcción de hiperplanos en un espacio multidimensional que separa casos de muestras etiquetadas de diferentes clases (Carlos M. Travieso, 2008). De manera más formal, una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación (o regresión). De este modo, una buena separación entre las clases permitirá una clasificación correcta (Gavidia Bovadilla, 2012)

La formulación matemática de las Máquinas de Vectores Soporte varía dependiendo de la naturaleza de los datos; es decir, existe una formulación para los casos lineales y, por otro lado, una formulación para casos no lineales.

3.2.6.2 Caso linealmente separable

Las MVS conforman hiperplanos que separan los datos de entrada en dos subgrupos que poseen una etiqueta propia. En medio de todos los posibles planos de separación de las dos clases etiquetadas como $\{-1, +1\}$, existe sólo un hiperplano de separación óptimo, de forma que la distancia entre el hiperplano óptimo y el valor de entrada más cercano sea máxima (maximización del margen) con la intención de forzar la generalización de la máquina que se esté construyendo. (Colmenares Lacruz, 2009)

Definición 1. *Un conjunto de vectores* $\{(x_1, y_1), \dots, (x_n, y_n)\}$ *donde* $x_i \in \mathbb{R}^d$ *e* $y_i \in \{-1,1\}$ *para* $i = 1, \dots, n$ *se dice separable si existe algún hiperplano en* $\mathbb{R}^d$ *que separa los vectores*

$X = \{x_1, \dots, x_n\}$ *con etiqueta* $y_i = 1$ *de aquellos con etiqueta* $y_i = -1$. (L. Gonzales)

Dado un conjunto separable existe (al menos) un hiperplano

$$\pi: \omega . x + b = 0$$

que separa los vectores $x_{i,}$ $i = 1, \dots, n$. Ver Figura 20.

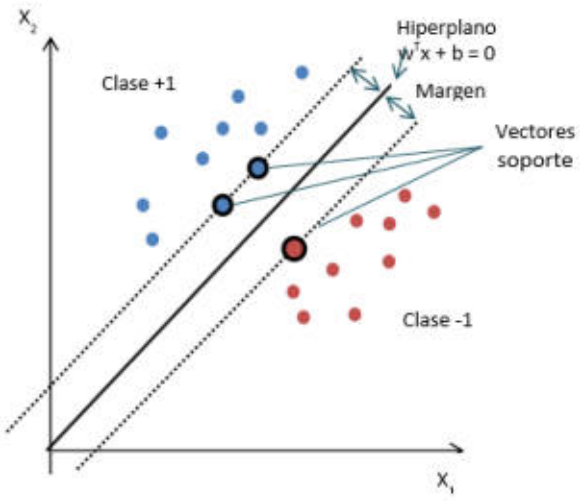

Figura 20 — Representación del SVM con sus elementos

Por tanto ahora se optimizará el margen

$$\min_{w \epsilon \mathbb{R}^d} \frac{1}{2} \|w\|^2$$

Las SVMs buscan entre todos los hiperplanos separadores uno que maximice la distancia de separación entre los conjuntos $\{(x_i, 1)\}$ y $\{(x_i, -1)\}$ (las dos clases posibles). Veamos el planteamiento del problema de optimización: cuando se fija un hiperplano separador es posible siempre reescalar los parámetros w y b de modo que:

$$x_i \, . w + b \geq +1 \, para \, y_i = +1 \;\; (Region \, A)$$

$$x_i \, . w + b \leq -1 \, para \, y_i = -1 \;\; (Region \, B)$$

De esta forma la unidad es la mínima separación entre el hiperplano separador y los vectores; y las dos desigualdades se pueden enunciar en una sola:

$$y_i(x_i.w + b) - 1 \geq 0, \qquad i = 1, \dots, n$$

Entonces, se obtiene el problema de optimización de las Máquinas de Vectores Soporte:

$$\min_{w \epsilon \mathbb{R}^d} \frac{1}{2} \|w\|^2$$

$$sujeto\ a\ \ y_i(x_i.w + b) \geq 1, \ \ \forall_i$$

Tambien denominado problema de optimización cuadrático con restricciones. Estos problemas son tratados introduciendo el método de multiplicadores de Lagrange (Morales España, y otros, 2005).

$$L_P(\boldsymbol{w}, b, \ \alpha_i) = \frac{1}{2} \ \|\boldsymbol{w}\ \|^2 - \sum_{i=1}^{l} \alpha i \ [yi(\boldsymbol{w}.\boldsymbol{x}_i \ + \ b) - \ 1]$$

donde L es minimizada sobre **w** y b, entonces calculamos $\frac{\partial L}{\boldsymbol{w}}$ y $\frac{\partial L}{b}$ e igualamos a cero, obteniendo las ecuaciones siguientes:

$$\frac{\partial L(\boldsymbol{w},b,\alpha)}{\boldsymbol{w}} \ = \ \boldsymbol{w} \ - \ \sum_{i=1}^{n} y_i \alpha_i \boldsymbol{x}_i = 0$$

$$\frac{\partial L(\boldsymbol{w},b,\alpha)}{b} \ = \ \sum_{i=1}^{n} y_i \alpha_i = 0$$

que al sustituir en la ecuación de Lagrangiano dará (Gala García , 2013):

$$L(w, b, \alpha) = \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j x_i.x_j - \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j x_i.x_j =$$
$$\sum_{i=1}^{l} \alpha_i y_i b + \sum_{i=1}^{l} \alpha_i = \ \theta(\alpha)$$

Entonces, el problema dual para las SVM es:

$$\max_{\alpha} \ \theta(\alpha) \ = \ \sum_i \alpha_i \ - \ \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i.\boldsymbol{x}_j$$

$$\alpha_i \ \geq \ 0 \ \forall_i$$

$$\sum_i \alpha_i y_i = 0$$

Finalmente, la condición KKT complementaria es:

$$\alpha_i^*(y_i((w \ \cdot \ x_i \ + \ b^*) \ - \ 1)) \ = \ 0$$

Ya obtenida la óptima solución del problema dual de la SVM ($a^* * b^*$), de la condición KKT y las derivadas parciales del Lagrangiano, se puede obtener los pesos $\boldsymbol{w}^*$ del problema primal a traves de la ecuación $\frac{\partial L}{\boldsymbol{w}} = 0$:

$$\boldsymbol{w}^* = \sum_{i=1}^{l} y_i \alpha_i x_i$$

Entre tanto, el término de bias corresponderá con:

$$b^* = y_i - w^* \cdot x_i = y_i - \sum_{i=1}^{l} \alpha_i^* y_i x_i . x_j$$

Asimismo, observamos de la condición KKT complementaria que si:

$\alpha_i^* > 0$, entonces:

$$y_i(\boldsymbol{w}.\boldsymbol{x}_i + b^*) = 1$$

Por consiguiente, estos son los puntos que estarán en el hiperplano. Resumiendo, por las ecuaciones se deduce que los puntos que están definiendo los pesos son aquellos que están en el hiperplano óptimo, en los que $\alpha_i^* > 0$, de ahí se tiene la condición última. A estos puntos los denominaremos vectores soporte.

En conclusión, el hiperplano óptimo se representará mediante α^* y b^* como (Gala García , 2013):

$$f(x, w^*, b^*) = \sum_{i \epsilon vs} y_i \alpha_i x_i \cdot x + b^*$$

3.2.6.3 Caso no linealmente separable

En la realidad, muchos de los problemas no tienen datos linealmente separables, lo que dificulta la tarea de encontrar un hiperplano que separe perfectamente los datos. (Gala García , 2013)

En el caso no lineal es necesario mencionar dos casos (Colmenares Lacruz, 2009):

- El primero: Se da cuando los datos pueden ser separables con margen máximo pero en un espacio de características (que es de una mayor dimensión, obtenido mediante una transformación a las variables del espacio de entrada) a través de la utilización de una función kernel.

- El segundo: También llamado "Soft Margin" o "Margen Blando", es usado cuando no es posible transformar los datos para separarlos linealmente, sea en el espacio de entrada o en el espacio de características.

3.2.6.4 Kernels

Hallar un hiperplano de óptima separación en el espacio de características es frecuentemente complicado y con un alto costo computacional. (Condori Arias, 2013)

- **Función de núcleo:** Siendo, el espacio de entrada *X*, el de características dotadas de un producto interno *H,* con una función $F : X \rightarrow H$, con *H* espacio inducido de Hilbert (Gala García , 2013), la función de núcleo $K : X \times X \rightarrow R$ se define como:

$$K(x_i , x_j) = \phi(x_i) \cdot \phi(x_j)$$

Al usar esta técnica encontramos un problema: que cualquier función no puede ser utilizada como núcleo, porque no es posible hallar una función *K* tal que:

$$K(x, y) = \phi(x) \cdot \phi(y)$$

Es fundamental, porque se puede construir hiperplanos óptimos en el espacio de características a través del uso del kernel sin la necesidad de tener en cuenta el espacio de características por sí mismo de manera explícita (Condori Arias, 2013).

Con el teorema de Mercer, tendremos las condiciones elementales para llevar a cabo lo anterior.

- **Teorema de Mercer:** Si una función escalar $k(x_i, x_j)$ es semidefinida positiva, existe una función $\Phi: \mathbb{R}^d \rightarrow F$, con F espacio de Hilbert, tal que $k(x_i, x_j)$ puede descomponerse como un producto escalar (Gala García , 2013):

$$k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

Por tanto, necesitamos dar las definiciones de matriz y kernel semidefinidos positivos.

Por consiguiente, requerimos proporcionar las definiciones de matriz semidefinida positiva y kernel semidefinidos positivos.

- **Matriz semidefinida positiva:** cuando es una matriz simétrica de tamaño n x n, y cumple que, para todo vector $x \in \mathbb{R}^n$ y $x \neq 0$, tenemos que:

$$x^T Kx \geq 0$$

- **Kernel semidefinido positivo:** Es semidefinido positivo cuando una función de kernel $k(x_i, x_j)$ cumple que, para cualquier otra función $f \in L^2$tenemos que:

$$\int k(x_i, x_j) f(x_i) f(x_j) dx_i dx_j$$

a) Tipos de Kernel

Los kernels mas utilizados son (Pedroza Bernal, 2007):

- **Kernel Lineal**: Su expresión es sencilla:

$$k(x_i, x_j) = x_i \cdot x_j$$

- **Kernel Gaussiano:** Se encarga de proyectar los vectores a un espacio de dimensión infinita y es expresado como:

$$k(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}$$

- **Kernel Polinomial:** Está asociada a un polinomio con coeficientes ai de propiedad conmutativa, se expresa como:

$$K(x_i, x_j) = (\gamma x_i . x_j + r)^d, \quad \gamma > 0; \quad r, d \in \mathbb{R}$$

- **Kernel de Base Radial:** (Radial Basis Function, RBF), se expresa como:

$$K(x_i, x_j) = c\ exp\left(-\gamma \|x_i - x_j\|^2\right), \quad \gamma > 0, \quad c \in \mathbb{R}$$

- **Kernel Sigmoide:** es un tipo de función que modela muchos procesos naturales y curvas de aprendizaje, se expresa como

$$K(x_i, x_j) = \tanh(\gamma x_i . x_j + r), \quad \gamma, r \in \mathbb{R}$$

3.2.6.5 Clasificador con margen suave y kernels (Condori Arias, 2013)

La expansión del kernel $K(x, x_i)$ nos posibilita construir una superficie de decisión en el espacio de entrada que es no lineal, sin embargo su imagen en el espacio de características sí es lineal. Tenemos la posibilidad de proponer la manera dual del problema de optimización con restricciones de un SVM, así:

Siendo el conjunto de entrenamiento $\{(x_i, t_i)\}_{i=1}^N$, encontraremos los multiplicadores de Lagrange $\{\alpha i\}$=1 N que maximice la función objetivo:

$$LD = \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j t_i t_j K(x_i, x_j)$$

sujeto a las restricciones:

$$\sum_{i=1}^N \alpha_i t_i = 0$$

$$0 \leq \alpha_i \leq C, \quad para\ i = 1,2,\dots,N$$

donde C es un parámetro positivo ajustable.

El problema dual que se termina de explicar es como para la situación de un hiperplano con margen suave, con excepción de que el producto interno $x_i^T x_j$ ha sido reemplazado por el $K(x_i, x_j)$.

3.2.6.6 Hipersuperficie de decisión (Condori Arias, 2013)

Sea la transformación no lineal Φ a partir del espacio de entrada, de dimensión m_0, al espacio de características, definiremos el hiperplano de decisión en el espacio de características de la siguiente manera:

$$\sum_{j=1}^{m_1} w_j \Phi_j(\boldsymbol{x}) + b = 0$$

donde $\{w_j\}_{j=1}^{m_1}$ representa el conjunto de pesos que enlaza el espacio de características con el espacio de salida, la dimensión del espacio de características es el valor de m_1 y el bias es b. De esta manera, se simplifica la ecuación anterior como:

$$\sum_{j=0}^{m_1} w_j \Phi_j(\boldsymbol{x}) = 0$$

donde $\Phi_0(\boldsymbol{x}) = 1$ para todo $\boldsymbol{x}$, por tanto el bias b es denotado por w_0.

Teniendo en cuenta la "imagen" proyectada en el espacio de características de los vectores de entrada $\boldsymbol{x}$, definiremos el hiperplano de decisión con la forma de:

$$\boldsymbol{w}^T \boldsymbol{\Phi}(\boldsymbol{x}) = 0$$

Adaptamos a la presente situación la ecuación (54), involucrando el espacio de características:

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{\Phi}(\boldsymbol{x}_i)$$

3.2.6.7 Propiedades de las SVM

Según Gala García (2013) y Martínez Ruedas (2006) las principales propiedades y ventajas de la utilización de las Máquinas de Soprte Vectorial son las siguientes:

- Cuando el modelo se encuentra bien parametrizado tiene una buena generalización con nuevos datos.
- El proceso de entrenamiento no es dependiente del número de atributos.

- La metamodelización es más fácil, porque los modelos dependen de pocos parámetros C, σ, ϵ.
- El mínimo es único por ser un problema de optimización convexa cuadrática.
- El modelo final viene a ser una combinación de unos pocos vectores soporte, que resulta sencillo.
- El entrenamiento de una SVM es esencialmente un problema de programación convexa cuadrática, que resulta interesante por dos razones:
  - Primero: Su eficiente computación.
  - Segundo: La garantía de encontrar un extremo global de la superficie de error, la solución resultante es la más óptima y única para los datos de entrenamiento dados.
- A la vez que minimiza el error de clasificación en el entrenamiento, maximiza el margen para mejorar la generalización del clasificador.
- No tiene el problema de Overfitting (Sobreentrenamiento) como podría ocurrir en las Redes Neuronales.
- La solución no es dependiente de la estructura del planteamiento del problema.
- Genera funciones no lineales, por medio de kernel. Trabaja con relaciones no lineales entre los datos. No es necesario trabajar en el espacio extendido, porque el producto escalar de los vectores transformados se puede sustituir por el kernel.
- Con pocas muestras de entrenamiento generaliza muy bien.

3.2.7 Identificación

Consiste en asociar la voz de un nuevo individuo, presentado al sistema, con alguna de las voces previamente registradas dentro del mismo (Martínez Mascorro, y otros, 2012).

Es un proceso por el que se determina a quien pertenece la muestra anónima aportada de entre un número de muestras registradas pertenecientes a distintos hablantes, es decir, a partir de un segmento de voz desconocido

tenemos que intentar descubrir qué persona es la que está hablando (Bonomo Laynez, 2012).

3.2.8 Métricas

La construcción de un método de clasificación en aprendizaje automático necesita la aplicación de métricas que permitan evaluar los resultados obtenidos.

3.2.8.1 Accuracy

El Accuracy es una métrica para evaluar modelos de clasificación. Informalmente, el **accuracy** es la fracción de predicciones que el modelo realizó correctamente. Formalmente, la Accuracy tiene la siguiente definición: (Google Developers, 2019)

$$Accuracy = \frac{Número\ de\ predicciones\ correctas}{Número\ total\ de\ predicciones}$$

3.3 Marco Conceptual

a) **Identificación de voz.** La identificación de voz se basa en intentar descubrir qué persona es la que está hablando.

b) **La tasa de muestreo.** Es la cantidad de muestras que se tomará en un intervalo de tiempo para convertir la señal analógica en señal digital.

c) **Profundidad de bit.** La profundidad de bit es el número de bits utilizados para almacenar cada muestra de una señal analógica.

d) **PCM.** Es un procedimiento de modulación utilizado para transformar una señal analógica en una señal digital.

e) **Máquinas de vectores soporte.** Un SVM es un clasificador binario que modela la frontera la decisión entre dos clases mediante un hiperplano separador. El hiperplano puede ser lineal y no lineal. En el caso de datos que no se pueden separarse por una frontera lineal en el espacio de características, dichos datos se transforman a otro espacio de características de mayor dimensión, por medio de una función kernel. En el nuevo espacio, las dos clases sí pueden ser separadas por una frontera lineal.

f) **Modelos de mezclas gaussianas.** El modelo de mezclas gaussianas abreviado como GMM, es un modelo probabilístico en el cual se hace la suposición de que todos los datos son generados de distribuciones gaussianas de mezcla finita con parámetros desconocidos.

g) **Modelos Ocultos de Markov.** Los Modelos Ocultos de Markov son modelos matemáticos de procesos estocásticos, procesos que generan secuencias aleatorias de salida de acuerdo a ciertas probabilidades.

h) **Accuracy.** La Accuracy es el porcentaje de acierto de las predicciones de un modelo de clasificación, siendo una de las métricas para evaluar diversos modelos de clasificación.

CAPÍTULO IV

METODOLOGÍA

4.1 Tipo y nivel de investigación

a) Tipo de investigación

En esta elaboración y desarrollo de la presente tesis se utilizó la investigación aplicada porque se utilizan los conocimientos adquiridos para que luego de investigar se obtengan otros conocimientos. Para Ñaupas Paitán y otros (2018) es aquella que basándose en los resultados de la investigación básica, pura o fundamental está orientada a resolver los problemas de la vida social de la comunidad regional o del país.

b) Nivel de investigación

El nivel de investigación que determinamos fue experimental. Para Hernández Sampieri y otros (2000) es la situación de control en la cual se manipulan, de manera intencional, una o más variables independientes (causas) para analizar las consecuencias de tal manipulación sobre una o mas variables dependientes (efectos).

4.2 Diseño de la investigación

Se utilizó el **diseño experimental.** Para Hernández Sampieri y otros (2000), en los diseños experimentales se manipulan, de manera intencional, una o más variables independientes (causas) para analizar las consecuencias de tal manipulación sobre una o más variables dependientes (efectos).

Tabla 2 — Diseño experimental de la investigación

Docentes	Modelos de Clasificación		
	HMM	GMM	SVM
1	100	100	100
2	100	100	66.67
3	100	100	55.56
. . .	. . .	. . .	. . .
31	95.70	97.85	29.47
32	94.79	95.83	30.21

4.3 Población y muestra

La población estuvo compuesta por 128 docentes de la UNAMBA sede central.

$$N = 128$$

El método utilizado para la muestra es el Muestreo Aleatorio Estratificado (MAE), porque la población fue divida en estratos y brindó las mismas oportunidades de ser seleccionadas a todas las personas que la componen.

Fórmula para el tamaño de la muestra:

$$n = \frac{NZ^2pq}{(N-1)\varepsilon^2+Z^2pq}$$

Donde:

N: Tamaño de la población.

Z: Nivel de confianza.

ε : Error máximo permitido.

p : Probabilidad de éxito o proporción esperada.

q : Probabilidad de fracaso.

Entonces, el tamaño de muestra para la investigación es:

$$n = \frac{128(1.28)^2(0.5)(0.5)}{(128-1)(0.1)^2+(1.28)^2(0.5)(0.5)} = 32$$

La población que son los 128 docentes de la UNAMBA sede central, fueron estratificadas en relación a las 04 facultades existentes en la UNAMBA, de esta manera se puede decir que se trabajó con 04 estratos.

Fórmula para el tamaño del muestreo estratificado:

$$n_i = \frac{N_i}{N} x\, n\,, \quad \forall\, i > 1$$

Donde:

i = número de estratos.

Aplicando la fórmula del muestreo estratificado a la investigación:

- Tamaño de muestra para la Facultad de Administración:

$$n_1 = \frac{N_1}{N} x\ n = \frac{20}{128} x\ 32 = 5$$

- Tamaño de muestra para la Facultad de Educación y Ciencias Sociales:

$$n_2 = \frac{N_2}{N} x\ n = \frac{18}{128} x\ 32 = 4{,}5$$

- Tamaño de muestra para la Facultad de Ingenierías:

$$n_3 = \frac{N_3}{N} x\ n = \frac{72}{128} x\ 32 = 18$$

- Tamaño de muestra para la Facultad de Medicina, Veterinaria y Zootecnia:

$$n_4 = \frac{N_4}{N} x\ n = \frac{18}{128} x\ 32 = 4{,}5$$

Tabla 3 — Aplicación de muestreo aleatorio estratificado

FACULTAD	Número de Docentes	Aplicando el MAE	Tamaño de Muestra "n"
Administración	20	5	5
Educación y Ciencias Sociales	18	4,5	4
Ingeniería	72	18	18
Medicina, Veterinaria y Zootecnia	18	4,5	5
TOTAL	128	32	32

4.4 Procedimiento

I Etapa: Exploración de los métodos de clasificación

En esta etapa se desarrolló la exploración de los métodos de clasificación.

- Búsqueda de Información.
- Análisis del pre-procesamienro y extracción de caracteristicas.
- Análisis de los métodos de clasificación.

II Etapa: Codificación de los métodos de clasificación

En esta etapa, se codificó en el lenguaje de programación Python cada uno de los métodos de clasificación, utilizándose las siguientes librerías:

- Para los Modelos Ocultos de Markov (HMM), se utilizó la librería hmmlearn.
- Para los Modelos de Mezclas Gaussianas (GMM), se utilizó la librería sklearn con el paquete mixture.
- Para los Máquinas de Vectores Soporte (SVM), se utilizó la librería sklearn con el paquete svm.

III Etapa: Procesamiento de datos

En esta etapa de procesamiento de los datos se realizó un registro de la información obtenida del prototipo de cada método de clasificación:

- Registro de la información del prototipo de consola correspondiente al método: Modelos Ocultos de Markov (HMM).
- Registro de la información del prototipo de consola correspondiente al método: Modelos de Mezclas Gaussianas (GMM).
- Registro de la información del prototipo de consola correspondiente al método: Máquinas de Vectores Soporte (SVM).

IV Etapa: Tratamiento de datos

En esta etapa se realizó un tratamiento de los datos tomados con los diferentes métodos, comparando el nivel de identificación del locutor.

- Comparación de los métodos y formulación de los resultados.

4.5 Técnicas e instrumentos

a) Técnicas

Los datos se recolectaron mediante:

- Grabación de audios.

b) Instrumentos

- Celular y surface.
- Registro de audios.
- Registro de Accuracy.

4.6 Análisis estadístico

Para el tratamiento de datos, se efectuó una serie de análisis que se exponen a continuación:

a) Análisis descriptivos. Mediante los cuales se observa las puntuaciones medias obtenidas de cada método y figuras para una mejor explicación.

Por el tipo de estudio, nuestra población y muestra fueron los docentes de la UNAMBA sede central, por lo tanto se realizó la prueba de Diseño por Bloques Aleatorizados: Muestras experimentales para determinar el mejor método de la investigación.

CAPÍTULO V

RESULTADOS Y DISCUSIONES

En este capítulo se muestran los experimentos y resultados obtenidos de este trabajo de tesis. Describiendo la arquitectura, el conjunto de datos y aplicando la métrica de Accuracy en los resultados obtenidos.

5.1 Análisis de resultados

5.1.1 Captura de voz

Para la realizacion de captura de voz por cada docente de la UNAMBA se utilizó el software Audacity y/o app grabador de audio, la grabación se realizó con un entorno de ruido en los ambientes de la UNAMBA. La grabación y segmentación de los archivos de audio se realizó según la técnica sugerida por Colaboradores de VoxForge (2019) en "Cómo segmentar manualmente un libro de audio (borrador)", la lectura del texto a grabar fue uno de los textos sugeridos por VoxForge (se puede ver en la sección Anexos-4) y otros textos aleatorios, en idioma castellano, guardando la grabación en un archivo de audio con formato "wav", modulación PCM y frecuencia de muestreo de 44100 Hz, con una duración mayor a 1 minuto por cada docente. Para la creación de un conjunto de datos de voz propuesto por Voxforge, se realiza la segmentación del archivo de audio grande en archivos de audio mas pequeños de 5 segundos a 10 segundos, basados en las pausas o silencios en la lectura de voz.

Como podemos ver, para la realización de este paso utilizamos como referencia las técnicas sugeridas por VoxForge, ya que es un conjunto de datos de voz muy utilizado en investigaciones recientes orientadas a la identificación y reconocimiento de voz. El conjunto de datos de VoxForge consta de muestras de voz multilingües donadas por la gente, mediante la lectura de unas frases. Este conjunto de datos contiene muestras de voz de 5 segundos, dado que las muestras de voz las graban los usuarios con su propio microteléfonos, la calidad varía significativamente entre diferentes

muestras. Este conjunto de datos contiene 25420 muestras en inglés, 4021 muestras francesas y 2963 muestras alemanas.

Tabla 4 — Características de cada archivo de audio

Parámetros del audio digital	
Tasa de muestreo	44100hz
Resolución bit	16 bits
Formato de audio	*.wav
Tipo de modulación	PCM

5.1.2 División del conjunto de datos

Se estableció la división de datos basándonos en Elkan (2012) el cual nos recomienda como regla general lo siguiente: para los datos del entrenamiento un 70% del total, y para los datos del test el 30% restante. La división de datos se realizó de forma aleatoria y estratificada, asegurando una distribución proporcional en la cantidad de los datos de voz para los conjuntos de entrenamiento y prueba.

5.1.3 Conjunto de datos de docentes

En esta investigación se seleccionó una muestra de 32 docentes y se decidió hacer un conjunto de datos con las voces de los docentes de la UNAMBA, etiquetando y numerando las voces de los 32 docentes, con 10 archivos de audios por docente, haciendo un total de 320 archivos de audio a tratar en esta experimentación.

En la Tabla 4, se visualiza la distribución de los datos, estos se dividen en 2 fases: el entrenamiento y el test. Se estableció la cantidad de 10 audios por cada docente, siendo el número de 7 archivos de audios para el entrenamiento y 3 archivos de audios para el test. Haciendo un total de 224 archivos de audios para el entrenamiento y 96 archivos de audios para el test, esto es para 32 docentes.

Tabla 5 — Tamaño del conjunto de datos usado en el experimento

POR DOCENTE (1)		TOTAL DE DOCENTES (32)	
Fases	**Nro. de audios**	**Fases**	**Nro. de audios**
Entrenamiento	7	Entrenamiento	224
Test	3	Test	96
Total	**10**	**Total**	**320**

5.1.4 Pre procesamiento

Despues de capturar la señal de voz en un audio, se procede a realizar este paso que consiste en:

1. La reducción de ruido (reducir el ruido de ambiente).
2. La normalización.
3. La detección de punto final (eliminar el silencio).

El pre procesamiento se realizó externamente con el software Audacity.

5.1.5 Extracción de características

Para la extracción de caracteristicas sobre el conjunto de datos se ha utilizado la librería de Python (Python Speech Features). El proceso de extracción de características de un archivo de audio se hace con una ventana de tamaño 25 ms, con un paso de 10 ms. Todos los archivos de audio se encuentran grabados a una frecuencia de muestreo de 44.1KHz, es decir, si tenemos un audio de 01 segundo de duración, esto supone que tendrá 44100 muestras. El tamaño de nuestro vector de características es 40, compuesto por 20 coeficientes MFCC y 20 coeficientes delta.

A continuación se muestra el código de la extracción de caracteristicas:

```
def  extract_features (audio, tasa):
    mfcc_feature = mfcc.mfcc (audio, tasa, 0.025 ,
    0.01 , 20 , nfft = 1200 , appendEnergy = True )
    mfcc_feature = preprocessing.scale (mfcc_feature)
    delta = calculate_delta (mfcc_feature)
    combinado = np.hstack ((mfcc_feature, delta))
    retorno combinado
```

5.1.6 Métodos de clasificación

Para construir el sistema de identificación de voz a partir de las características extraídas anteriormente, necesitamos modelar todos los archivos de audio de los docentes de forma independiente. Para ello, empleamos los siguientes metodos:

a) Los Modelos Ocultos de Markov (HMM).

- **Entrenamiento**

  Para entrenar los audios recopilados mediante el método HMM, se ejecuta una vez, aprendiendo las características de voz por cada audio, usando 7 archivos de audio por docente para obtener los parámetros que utilizaremos en el test.

  A continuación se muestra el código que se utiliza para entrenar los modelos de los docentes en HMM.

```
    if count == 7:
        model = hmm.GMMHMM(n_components=states_num,
                           n_mix=GMM_mix_num, \
                           transmat_prior=transmatPrior,
                           startprob_prior=startprobPrior, \
                           covariance_type='diag', n_iter=10)
        model.fit(features)
```

- **Test**

  En el Test del HMM, al ingresar un archivo de audio de prueba para la identificación de voz, se extrae los 40 vectores características para el mismo, luego se obtienen 2 probabilidades, una probabilidad parcial y otra probabilidad total. Finalmente, se da por ganador o como docente identificado a la probabilidad con mayor puntuación.

  A continuación se presenta el código que predice el docente del audio de prueba.

```
                vector = extract_features(audio, sample_rate)
                log_likelihood = np.zeros(len(models))
                for i in range(len(models)):
```

```
4                       hmm = models[i]
5                       scores = np.array(hmm.score(vector))
6                       log_likelihood[i] = scores.sum()
7               winner = np.argmax(log_likelihood)
8               print "\t detectado como ", speaker_names[winner]
```

b) Los Modelos Mezclas Gaussianas (GMM)

- **Entrenamiento**

  Para entrenar los modelos de docentes en GMM con 16 componentes, se ejecuta una vez aprendiendo de las características voz por cada audio, utilizando 7 archivos de audio por docente para obtener los parámetros que utilizaremos en el test.

  A continuación, se muestra el código que se utiliza para entrenar los modelos de los docentes en GMM.

```
1 if count == 7:
2         gmm = GMM(n_components =16, max_iter = 200,
3                   covariance_type='diag',n_init = 3)
4         gmm.fit(features)
```

- **Test**

  En el Test del GMM, al ingresar un archivo de audio para la identificación del hablante, se extrae los 40 vectores características para el mismo, luego se obtienen 2 probabilidades, una probabilidad parcial y otra probabilidad total. Finalmente, se da por ganador o como docente identificado a la probabilidad con mayor puntuación.

  A continuación se presenta el código que predice el docente del audio de prueba.

```
1         vector = extract_features(audio, sr)
2         log_likelihood = np.zeros(len(models))
3         for i in range(len(models)):
4             gmm = models[i]
5             scores = np.array(gmm.score(vector))
6             log_likelihood[i] = scores.sum()
```

```
7         winner = np.argmax(log_likelihood)
8         print("\t detectado como :", speakers[winner])
```

c) Máquinas De Vectores Soporte (SVM).

- **Extracción de características:**

  Para un archivo de audio genera Coeficientes de Mel, luego se calcula la media y la desviación estándar de los MFCC para ser utilizados como vector de características.

```
1        def _generar_features(self, data_dir, outfile):
2         with open(outfile,'w') as abrir:
3             melwriter = csv.writer(abrir)
4             speakers = os.listdir(data_dir)
5             for spkr_dir in speakers:
6                 for soundclip in os.listdir(os.path.join(data_dir,
7 spkr_dir)):
8                     clip_path =
9 os.path.abspath(os.path.join(data_dir, spkr_dir, soundclip))
10                    sample_rate, data = wavfile.read(clip_path)
11                    ceps = mfcc(data, sample_rate)
12                    fvec = self._mfcc_to_fvec(ceps)
13                    fvec.append(spkr_dir)
```

- **Entrenamiento**

  Para entrenar los modelos de docentes en el SVM, se utiliza la función “fit” de la clase “Reconocedor”, se ejecuta aprendiendo de las características de voz por cada audio, usando 7 archivos de audio por docente para obtener los parámetros que utilizaremos en el test.

  A continuación se muestra el código que se utiliza para entrenar los modelos de los docentes en GMM.

```
        .
        .
        .
        self.recognizer = svm.SVC()
        melv_list, speaker_names = self._getData(saved_file)
        .
        .
        .
        self.recognizer.fit(melv_list, speaker_ids)
```

- **Test**

  Para el Test en el SVM se utiliza la funcion **predict,** que se encarga de reconocer al locutor en el archivo de audio por medio del vector de características.

```
            speaker = recognizer.predict("DataTest4/"+speaker_item)
            print "Detectado como: %s \n" % speaker
```

```
    def predict(self, soundclip):
        sample_rate, data = wavfile.read(os.path.abspath(soundclip))
        ceps = mfcc(data, sample_rate)
        fvec = self._mfcc_to_fvec(ceps)
        speaker_id = self.recognizer.predict([fvec])[0]
```

5.1.7 Resultados de la métrica (Accuracy)

Despues del entrenamiento y la etapa de prueba se evaluan los resultados con la métrica de Accuracy, este experimento se realizó sobre una muestra de 32 docentes de la UNAMBA. El promedio de identificación que se obtuvo para el método Modelos Ocultos de Markov es del 94.79%, para el método Modelos de Mezclas Gaussianas es de 95.83%, para el método Máquinas de Vectores Soporte es del 30.21%, ver Tabla 6.

Tabla 6 — Accuracy de los métodos HMM, GMM y SVM

Nro. Docentes	HMM	GMM	SVM
32	94.79%	95.83%	30.21%

Entonces, determinamos que el mejor método de clasificación para la identificación de voz es el correspondiente a la mayor puntuación de Accuracy, siendo el método de clasificación: Modelos de Mezclas Gaussianas con una predicción de 95.83%, como se muestra en la Figura 21.

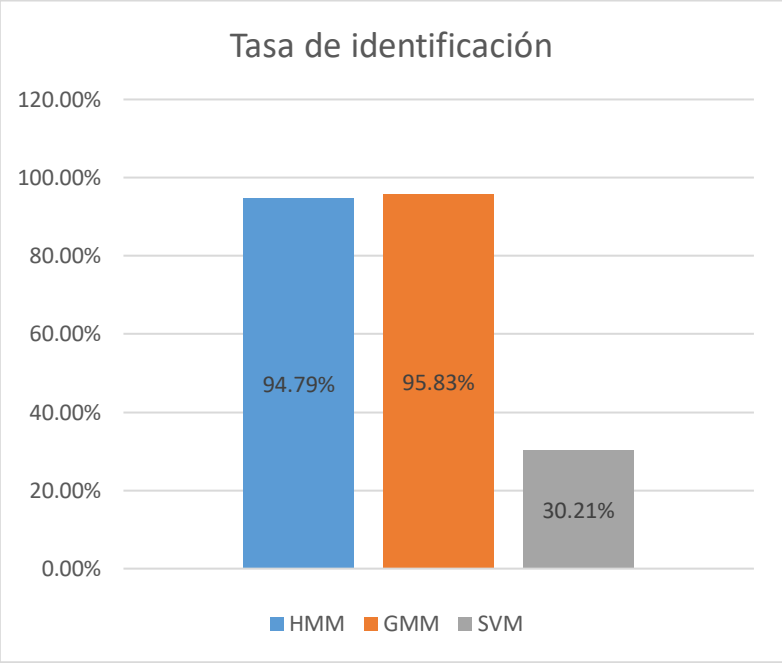

Figura 21 — Tasa de identificación de HMM, GMM, SVM

Estos resultados nos indican que a un incremento de docentes la predicción en la identificación de voz con los métodos GMM y HMM se reduce significativamente, entonces la identificación de voz rinde en poblaciones grandes, mientras que en el método SVM, a un incremento de docentes la identificación se reduce notablemente, que indica que la identificación de voz no rinde en poblaciones grandes, ver Figura 21.

5.2 Contrastación de hipótesis

5.2.1 Prueba de hipótesis general: Determinar el mejor método entre HMM, GMM y SVM.

a. Formulación de la hipótesis estadística

H0: $\mu_1=\mu_2=\mu_3$ [Todos los metodos de identificación de voz son iguales (Modelos Ocultos de Markov, Modelos de Mezclas Gaussianas y Máquinas de Vectores Soporte)].

H1: $\mu 1 \neq \mu 2 \neq \mu 3$ [Al menos dos métodos de identificación de voz son diferentes (Modelos Ocultos de Markov, Modelos de Mezclas Gaussianas y Máquinas de Vectores Soporte)]

b. Nivel de siginificancia

$$\propto = 5\% \cong 0.05$$

c. Estadístico de prueba

Se utiliza el ANOVA

Tabla 7 — Pruebas de los efectos inter-sujetos

Variable dependiente: Accuracy

Origen	Suma de cuadrados tipo III	gl	Media cuadrática	F	P
Modelo corregido	8,987[a]	2	4,494	570,305	,000
Intersección	56,941	1	56,941	7226,721	,000
Metodos	8,987	2	4,494	570,305	,000
Error	,733	93	,008		
Total	66,661	96			
Total corregida	9,720	95			

a. R cuadrado = ,925 (R cuadrado corregida = ,923)

Según la Tabla 7, se observa 96 tratamientos que se forman para combinar los niveles de los dos factores. El valor de P (probabilidad)

es equivalente a 0.00 y la combinación de Métodos por Accuracy ambos valores son menores al nivel de significancia $\alpha = 0.05$, por lo cual podemos concluir que se rechaza la hipótesis nula y se acepta la hipótesis alterna, también corroborado mediante la prueba F ($F_t<F_c$), siendo 3.011532167 < 570.305, por lo cual se concluye: Que al menos dos métodos de identificación de voz son diferentes (Modelos Ocultos de Markov, Modelos de Mezclas Gaussianas y Máquinas de Vectores Soporte).

Asi también se observa el valor de R cuadrado que es muy cercano a 1 (0.925), considerando altamente significativo la prueba.

Comparaciones en parejas de Tukey:

Tabla 8 — Tukey (Accuracy)

	N	Media	Desviación típica	Error típico	Intervalo de confianza para la media al 95%		Mínimo	Máximo
					Límite inferior	Límite superior		
GMM	32	,9900	,01185	,00210	,9857	,9942	,96	1,00
HMM	32	,9830	,02074	,00367	,9756	,9905	,95	1,00
SVM	32	,3375	,15188	,02685	,2827	,3922	,22	1,00
Total	96	,7702	,31987	,03265	,7053	,8350	,22	1,00

Según la Tabla 8, para la verificación del mejor método de identificación de voz se determinó tras usar la prueba de ANOVA y las comparaciones de Tukey, donde el Modelo de Mezclas Gaussianas tiene la media con un valor más alto de 0.99, por ellos se determina que este viene a ser el mejor método de clasificación.

Tabla 9 — Correlaciones

		GMM	HMM	SVM
GMM	Correlación de Pearson	1	,948**	,277
	Sig. (bilateral)		,000	,124
	Suma de cuadrados y productos cruzados	,004	,007	,015
	Covarianza	,000	,000	,000
	N	32	32	32
HMM	Correlación de Pearson	,948**	1	,266
	Sig. (bilateral)	,000		,141
	Suma de cuadrados y productos cruzados	,007	,013	,026
	Covarianza	,000	,000	,001
	N	32	32	32
SVM	Correlación de Pearson	,277	,266	1
	Sig. (bilateral)	,124	,141	
	Suma de cuadrados y productos cruzados	,015	,026	,715
	Covarianza	,000	,001	,023
	N	32	32	32

**. La correlación es significativa al nivel 0,01 (bilateral).

Según la Tabla 9, en el Coeficiente de Correlación de Pearson, se observa que el Método GMM y HMM es significativo al 0.01, con un R= 0.948, sin embargo el método SVM tiene un R=0.277 y R=0.266. En conclusión, el método SVM no tiene relación con los métodos GMM y HMM en la identificación de voz.

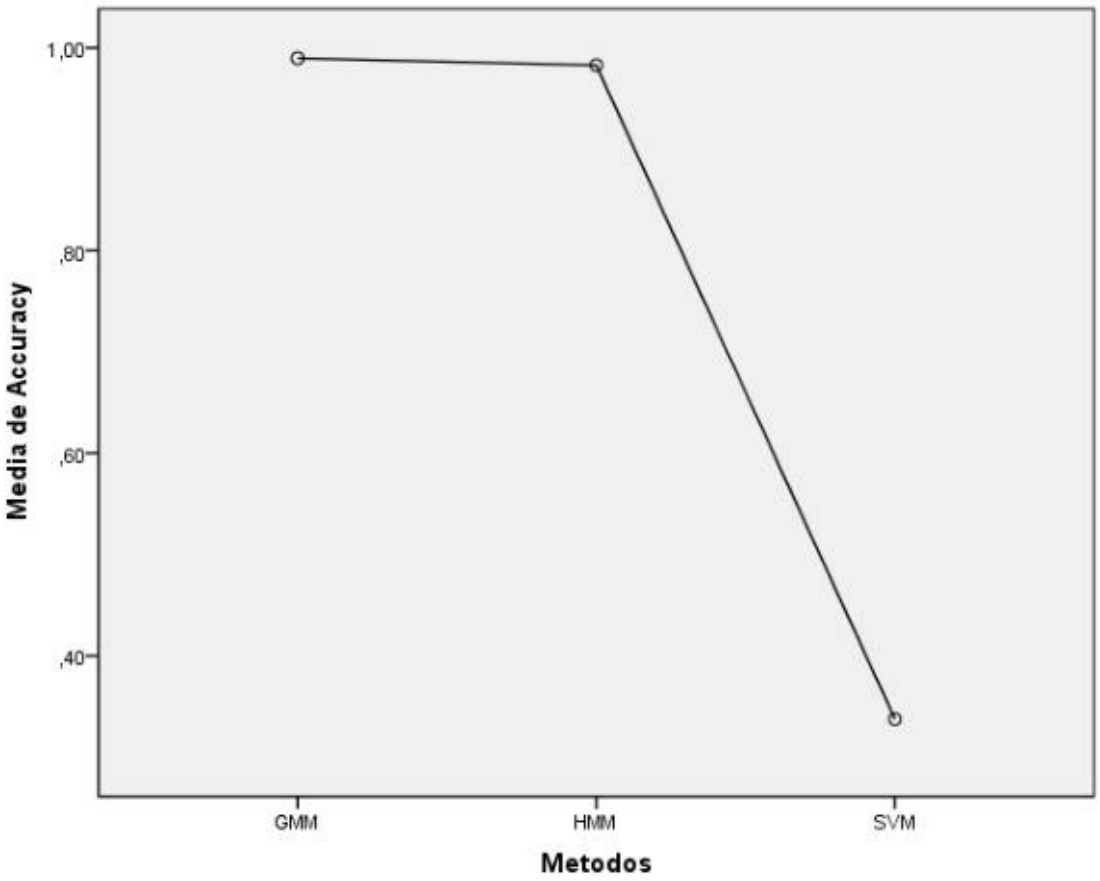

Figura 22 — Puntos de GMM, HMM y SVM

Extraído de Tabla 8

Según la Figura 22, determinamos que el mejor método son los Modelos de Mezclas Gaussianas. También, se aprecia que el método que ocupa el segundo lugar son los Modelos Ocultos de Markov y finalmente que tercer y último lugar encontramos a las Máquinas de Vectores Soporte.

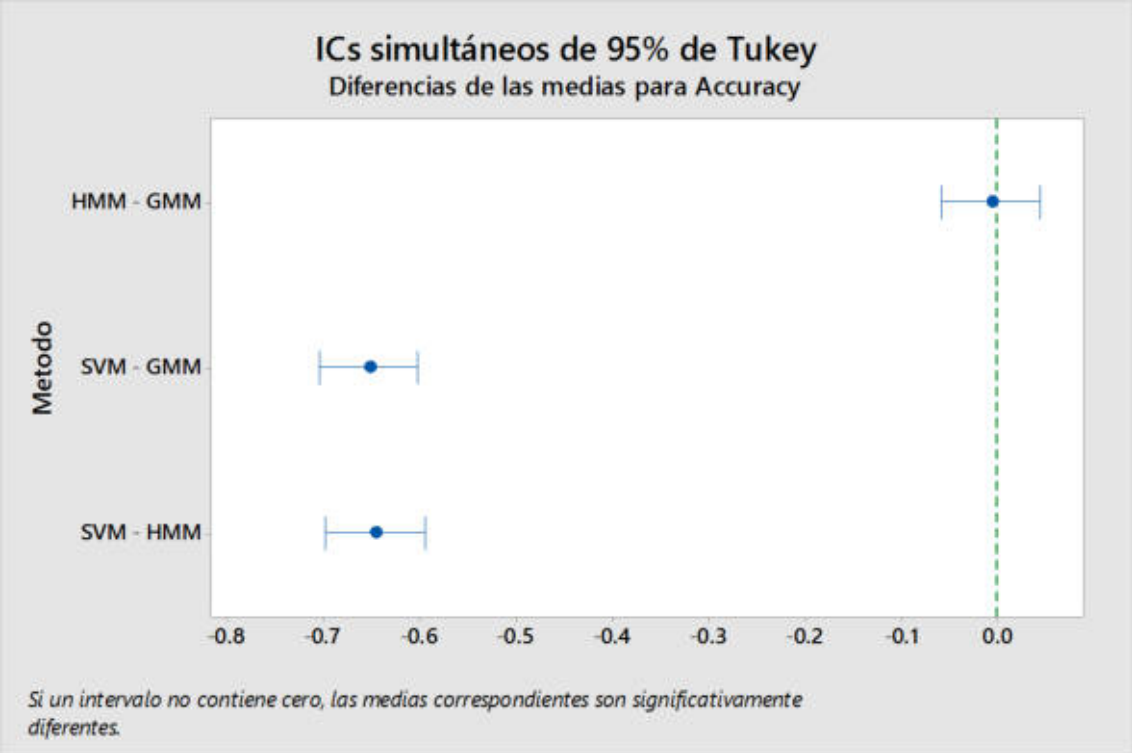

Figura 23 — ICs simultáneos de 95% de Tukey
Extraído de Tabla 8

En los resultados de Tukey, ver Figura 23, los intervalos de confianza indican lo siguiente:

- El intervalo de confianza para la diferencia entre las medias de las métodos SVM-GMM y SVM-HMM es de -0.7040 a -0.6009 y -0.6971 a -0.5940 respectivamente. En esos rango no se incluyen el cero, los cuales indican que la diferencia es estadísticamente significativa.
- El intervalo de confianza de los métodos HMM-GMM incluye el cero, lo que indica que las diferencias no son estadísticamente significativas.

d. Región crítica

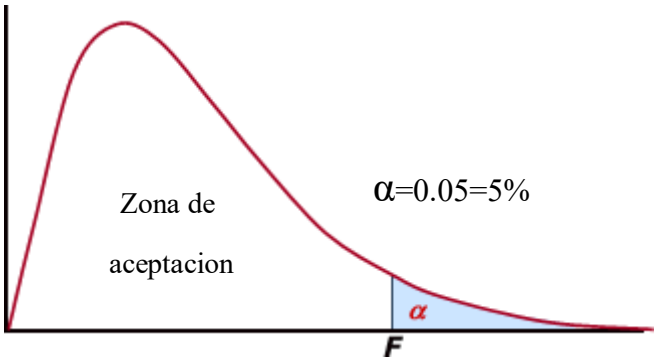

Figura 24 — Región Crítica

Si $\alpha \geq P_{obs}$ entonces se rechaza la H_0 y se acepta la hipótesis alterna.

e. Desición

Ya que, 0.5 > 0.0, se rechaza la H_0 y se acepta la hipótesis alterna, entonces se afirma que existen diferencias significativas entre las medias de los métodos GMM, HMM Y SVM.

Según la tabla 8, se observa que los Modelos de Mezclas Gaussianas tienen la media con el valor más alto, por lo que concluimos que el mejor método son los GMM, seguido por los HMM y el peor método son las SVM.

5.3 Discusión

Los GMM y los HMM son de un tipo de aprendizaje no supervisado, mientras que las SVM son de tipo supervisado, por lo que hacer una comparación entre estos en las mismas condiciones viene a ser una tarea muy dificil debido a la distinta orientacion de los modelos. Sin embargo, nuestros experimientos fueron realizados como si se tratara de un solo tipo de aprendizaje. Bajo las mismas condiciones se encontró que los modelos de aprendizaje no supervisado son superiores en efectividad (Accuracy) al modelo de aprendizaje supervisado (SVM). En otras palabras, los GMM y los HMM tienen mejor resultado que SVM, tal y como se muestra en los experimentos.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

Al concluir el trabajo de investigación se determinó lo siguiente:

- El mejor método estimado son los Modelos de Mezclas Gaussianas, porque dieron los mejores resultados para textos de audio de contenido independiente, con un Accuracy alcanzado de 95.83 %.
- Se implementó el conjunto de datos de voz con 10 archivos de audio por docente de un total de 32 docentes, obteniendo así 320 archivos de audio utilizados en la presente investigación.
- El método de los Modelos de Mezclas Gaussianas, al ser ejecutado dio buenos resultados para textos de audio de contenido independiente, alcanzado un Accuracy de 95.83 % en la identificación de voz, esto quiere decir que el Modelo de Mezclas Gaussianas realiza una buena clasificación de las características de la voz.
- El método de los Modelos Ocultos de Markov, al ser ejecutado también obtuvo buenos resultados, alcanzado un Accuracy de 94.79 % en la identificación de voz, muy próximo al mejor método determinado en esta investigación. Este método es eficiente porque predice la secuencia siguiente, pero a medida que incrementa la data tambien empieza a disminuir la predicción en la identificación de voz.
- El método de las Máquinas de Vectores Soporte, al ser ejecutado logró alcanzar un Accuracy de 30.21% en la identificación de voz. Este método no es muy utilizado para la clasificación de las características de la voz en la cual está enfocada la presente investigación, pero se logró implementar para un grupo reducido con buena predicción. El método de Máquinas de Vectores Soporte requiere de mucha más información sobre cómo debe ser utilizado en la identificación de voz.
- Se demuestra que aplicando el mismo pre-procesamiento y conjunto de datos en estos 3 métodos de clasificación es posible realizar la comparación con la métrica de evaluación denominado “Accuracy”, obtuviendo los siguientes

resultados: Para el método de los Modelos de Mezclas Gaussianas el 95.83 %, para el método de los Modelos Ocultos de Markov el 94.79% y para el método de las Máquinas de Vectores Soporte el 30.21%, determinando de esta manera que el mejor método de clasificación es el de los Modelos de Mezclas Gaussianas y el peor método de clasificación el de las Máquinas de Vectores Soporte.

6.2 Recomendaciones

- Despues de culminar el trabajo de investigación recomendamos utilizar los siguientes métodos: Los Modelos de Mezclas Gaussianas (GMM) y los Modelos Ocultos de Markov (HMM) para el desarrollo de software de identificación de voz, por los altos valores observados en la experimentación.
- También se recomienda hacer un estudio o investigación acerca de la identificación de emociones a través de la voz (alegria, tristeza, ira, calma), con los siguientes métodos: Los Modelos de Mezclas Gaussianas (GMM) y los Modelos Ocultos de Markov (HMM).
- Se recomienda que en investigaciones futuras se experimente con un conjunto de datos más grande.
- También se recomienda que para el desarrollo de software de identificación de voz se implemente las librerías de machine learning de Python, por los buenos resultados obtenidos en la presente investigación.
- Recomendamos usar una buena arquitectura de hardware para la clasificación y procesamiento de datos de voz.
- Recomendamos una investigación futura sobre las métricas de evaluación que se debe utilizar en los métodos de clasificación orientados a la identificación de locutor o identificación de voz.
- En el método de clasificación denominado las Máquinas de Vectores Soporte se recomienda investigar cuál es la forma de etiquetar los datos de voz para realizar un sistema de identificación de voz.

REFERENCIAS BIBLIOGRÁFICAS

1. **Aguilera Bonet, Pablo.** Reconocimiento de voz usando htk.

2. **Alagöz, Fatih. 2013.** Department of Computer Engineering. [En línea] Febrero de 2013. http://www2.cmpe.boun.edu.tr/courses/cmpe362/spring2014/files/projects/MFCC%20Feature%20Extraction.pdf.

3. **Auccapuma Gamarra, Jhon Dennis y Mamani Condori, Errol Wilderd. 2016.** *Identificación de locutor usando codebooks de coeficientes cepstrales en las frecuencias de Mel y modelos ocultos de Markov.* Cusco : s.n., 2016.

4. **Aya Gamal Osman Fatma A. Mohammed Ahmed Moawad Ahmed Helmy, Nes-ma Zein. 2012.** Smart Blind Stick. s.l. : Mansoura University, 2012.

5. **Bonomo Laynez, David. 2012.** Sistemas de verificación automática de locutor. Sevilla, Andalucía, España : s.n., Septiembre de 2012.

6. **Campos Rubio, Raúl. 2016.** Sistema para el reconocimiento de estrés en voz. España : s.n., 2016.

7. **Carlos M. Travieso, Jesús B. Alonso, Miguel A. Ferrer. 2008.** Reducción del vector de características en reconocimiento facial. 2008. pág. 4.

8. **Colaboradores de VoxForge. 2019.** Cómo segmentar manualmente un libro de audio (borrador). *VoxForge.* [En línea] VoxForge, 23 de Septiembre de 2019. [Citado el: 23 de Septiembre de 2019.] http://www.voxforge.org/home/dev/mansegaudio.

9. **Colmenares Lacruz, Gerardo A. 2009.** La web del profesor. *Pagina de Gerardo Colmenares.* [En línea] 03 de Diciembre de 2009. [Citado el: 20 de 10 de 2019.] http://webdelprofesor.ula.ve/economia/gcolmen/programa/economia/maquinas_vectores_soporte.pdf.

10. **Condori Arias, Elvis Franks. 2013.** Reconocimiento y clasificación de objetos usando inteligencia artificial basada en SVM y visión estereoscópica. Lima, Lima, Perú : s.n., 2013.

11. **Córdova Zamora, Manuel. 2006.** *Estadistica Aplicada.* Lima : Moshera S.R.L., 2006.

12. **E Tippens, Paul, González Ruíz, Angel Carlos y García Hernández, Ana Elizabeth. 2007.** *Física: conceptos y aplicaciones.* s.l. : McGraw-Hill Interamericana, 2007.

13. **Elkan, Charles. 2012.** Evaluating classifiers. California : San Diego: University of California, 2012.

14. **Esteve Elizalde, Cristina. 2007.** Reconocimiento de locutor dependiente de texto mediante adaptación de modelos ocultos de markov fonéticos. Madrid, Madrid, España : s.n., Julio de 2007.

15. **Fandiño Rodríguez, Deiby Alexander. 2005.** Estado del arte en el reconocimiento Automático de voz. Abril de 2005.

16. **Furui, Sadaoki. 1981.** Comparison of speaker recognition methods using statistical features and dynamic features. Tokio : IEEE, 1981. Vol. 29, 3.

17. **Gala García , Yvonne . 2013.** Algoritmos SVM para problemas sobre big data. Madrid, Madrid, España : s.n., 25 de Septiembre de 2013.

18. **García Herrero, Alberto. 2015.** Algoritmos para la estimación de modelos de mezclas gaussianas. Cantabria, Santander, España : s.n., Julio de 2015.

19. **Garretón Vender, Claudio Andrés. 2007.** Compensación no supervisada de variabilidad intra-locutor y ruido en reconocimiento de patrones de voz. Santiago de Chile, Santiago de Chile, Chile : s.n., Agosto de 2007.

20. **Gavidia Bovadilla, Giovana E. 2012.** Clasificadores Basados en Máquinas de Soporte Vectorial para el Diagnóstico y Predicción de la Enfermedad de Alzheimer. Barcelona, Barcelona, España : s.n., 2012.

21. **Gonzales Domínguez, Javier. 1998.** Nuevas Técnicas de compensación de canal en reconocimiento de locutor e idioma. Madrid : Universidad Autónoma de Madrid, 1998.

22. **Google Developers. 2019.** Machine Learning. *Machine Learning.* [En línea] 17 de Marzo de 2019. [Citado el: 17 de Marzo de 2019.]

https://developers.google.com/machine-learning/crash-course/classification/accuracy.

23. **Goutam, Saha, Sandipan, Chakroborty y Suman, Senapati. 2005.** A new silence removal and endpoint detection algorithm for speech and speaker recognition applications. *In Proceedings of the 11th National Conference on Communications (NCC).* 2005.

24. **Hansen, J.H, Proakis, J.G y Deller, J.R. 1987.** Discrete-time processing of speech signals. 1987.

25. **Harald Priewald, Robin. 2009.** Classification of Acoustic Plastic Pipe Water Leak Signals with Gaussian Mixture Models. 2009.

26. **Hernández García, Ruber. 2011.** *Detección de anuncios en emisiones de televisión basada en la caracterización del audio.* s.l. : Universidad de las Ciencias Informáticas, 2011.

27. **Hernández Sampieri, Roberto, Fernández Collado, Carlos y Baptista Lucio, María del Pilar. 2000.** *Metodología de la investigación.* [ed.] Pilar Baptista Lucio. Sexta. México : McGraw-Hill / Interamericana editores S.A. de C.V, 2000.

28. **Hub electronics. 2015.** Modulation and different types of modulation. [En línea] 28 de Agosto de 2015. [Citado el: 03 de Marzo de 2020.] https://www.electronicshub.org/modulation-and-different-types-of-modulation/.

29. **L. Gonzales, Abril.** Modelos de Clasificación basados en Máquinas de Vectores Soporte. Sevilla, Andalucía, España : s.n.

30. **Lawrence R., Rabiner. 1989.** A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE.* February de 1989. Vol. 77, 2.

31. **Martínez Mascorro, Guillermo Arturo y Aguilar Torres, Gualberto. 2012.** Sistema para identificación de hablantes robusto a cambios en la voz. Ecuador : Universidad Politécnica Salesiana, 2012. 8.

32. **Martínez Ruedas, Cristina. 2006.** Detección Multiusuario para DS-CDMA basado en SVM. Sevilla : Universidad de Sevilla, 2006.

33. **Morales España, Germán Andrés y Gómez Ruiz, Alvaro. 2005.** Estudio e implementación de una herramienta basada en máquinas de soporte vectorial aplicada a la localización de fallas en sistemas de distribución. Bucaramanga, Santander, España : s.n., 2005.

34. **Multison Online. 2016.** De analógico a digital: frecuencia de muestreo y tasa de bits. *Blog*. [En línea] Multison Online, 4 de Julio de 2016. [Citado el: 15 de Enero de 2021.] https://multisononline.com/blog/de-analogico-a-digital-frecuencia-de-muestreo-y-tasa-de-bits-n267.

35. **Networks at MIT group NETMIT. 2014.** Sparse fast fourier transfrom. [En línea] 2014. http://www.groups.csail.mit.edu/netmit/sFFT/algorithm.html.

36. **Ñaupas Paitán, Humberto , y otros. 2018.** *Metodología de la investigación cuantitativa-cualitativa y redacción de la tesis.* Quinta. Bogotá : Ediciones de la U, 2018.

37. **Ochoa, Felipe, San Martín, César y Carrillo, Roberto. 2008.** Identificación biométrica de locutores para el ámbito forense: Estado del arte. *VI Congreso Iberoamericano de Acústica-FIA 2008.* 2008.

38. **P. Campbell, Joseph . 1997.** Speaker Recognition: A tutorial. s.l. : Proceedings of the IEEE, 1997. Vol. 85, 9.

39. **Pedroza Bernal, Juan Gabriel. 2007.** Aplicación de las máquinas de soporte vectorial al reconocimiento de hablantes. 22 de Junio de 2007.

40. *Procesamiento de señales digitales: Informe final de reconocimiento de oradores.* **Zhou, Xinyu, Wu, Yuxin y Li, Tiezheng.**

41. **Rueda Rojo, Leticia. 2011.** Mejoras en reconocimiento del habla basadas en mejoras en la parametrización de la voz. 2011.

42. **Sayed Jaafer Abdallah, Izzeldin Mohamed Osman, Mohamed Elhafiz Mustafa. 2012.** Text-independent speaker identification using hidden markov model. *World of Computer Science and Information Technology Journal (WCSIT).* 2012.

43. **Sayed Jaafer, Abdallah, Izzeldin Mohamed , Osman y Mohamed Elhafiz, Mustafa. 2012.** Text-independent speaker identification using hidden markov model. 2012. Vol. 2, 6, págs. 203–208.

44. *Speaker identification using mfcc-domain support vector machine.* **Kamruzzaman , S. M, y otros. 2010.** 2010.

45. **Ugarte Echeverría, María Elísabe. 2010.** Implementación y comparación de algoritmos basados en técnicas biométricas de voz para reconocimiento de locutor en colaboración con la empresa IECISA. Pamplona : s.n., 30 de Junio de 2010.

46. **Velázquez López, Omar. 2017.** Implementación de técnicas de procesamiento para un transcriptor de música en tiempo real. Enero de 2017.

47. **Villalobos Vives , Sara y Yepes Lopez, Luis Alfredo. 2010.** *Modulacion por Impulsos Codificados "MIC o PCM".* Cartagena : Universidad Tecnológica de Bolívar, 2010.

48. *World of Computer Science and Information Technology Journal ISSN.* **2012.** 2012, World of Computer Science and Information Technology Journal ISSN.

49. **Zaso Candil, Rubén. 2014.** Análisis de compensación de variabilidad en reconocimiento de locutor aplicado a duraciones cortas. Julio de 2014.

ANEXOS

1. Codificación nativa de HMM

Inicialización de la ejecución de la codificación nativa de HMM. Seguidamente se ejecuta el entrenamiento en los HMM.

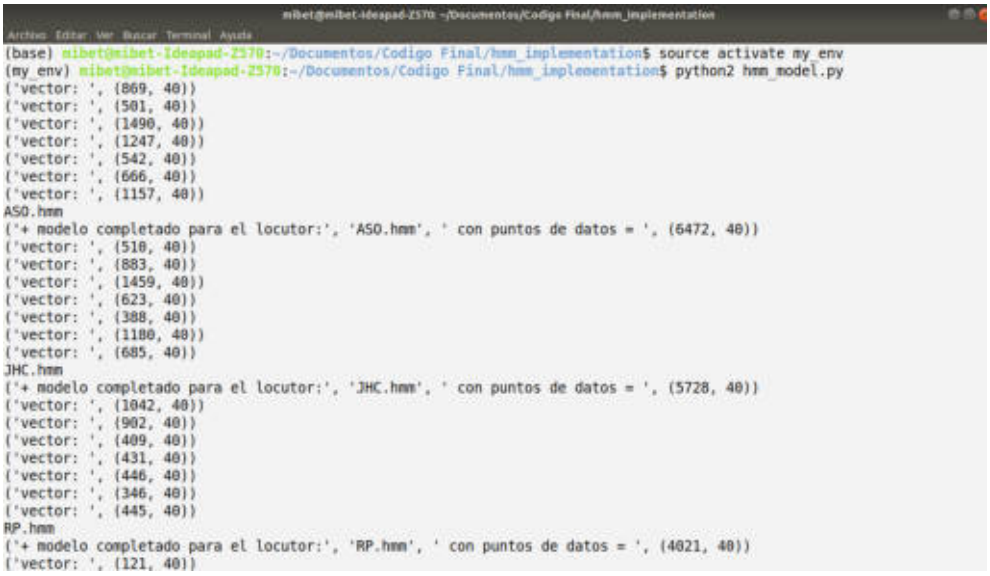

Figura A.1 — Entrenamiento del método HMM (32 docentes)

Datos de ejecución del entrenamiento del HMM, con 7 audios por docente. Luego ejecución del Test del HMM, cargando los 32 docentes entrenados.

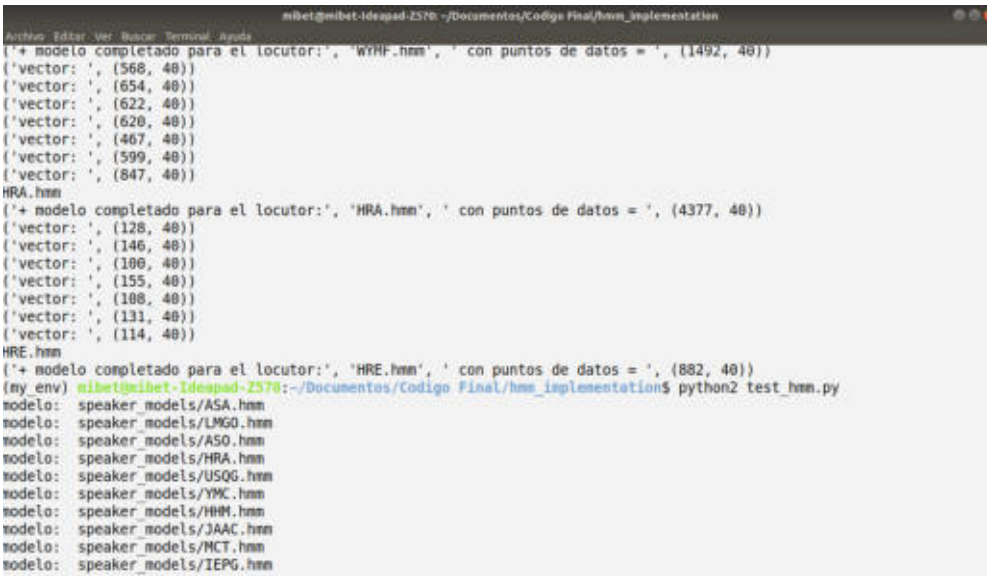

Figura A.2 — Carga de modelos entrenados con el método HMM (32 docentes)

Vista del Test, identificando a los 32 docentes, con 7 errores de 96 audios, dando un Accuracy de 92.708

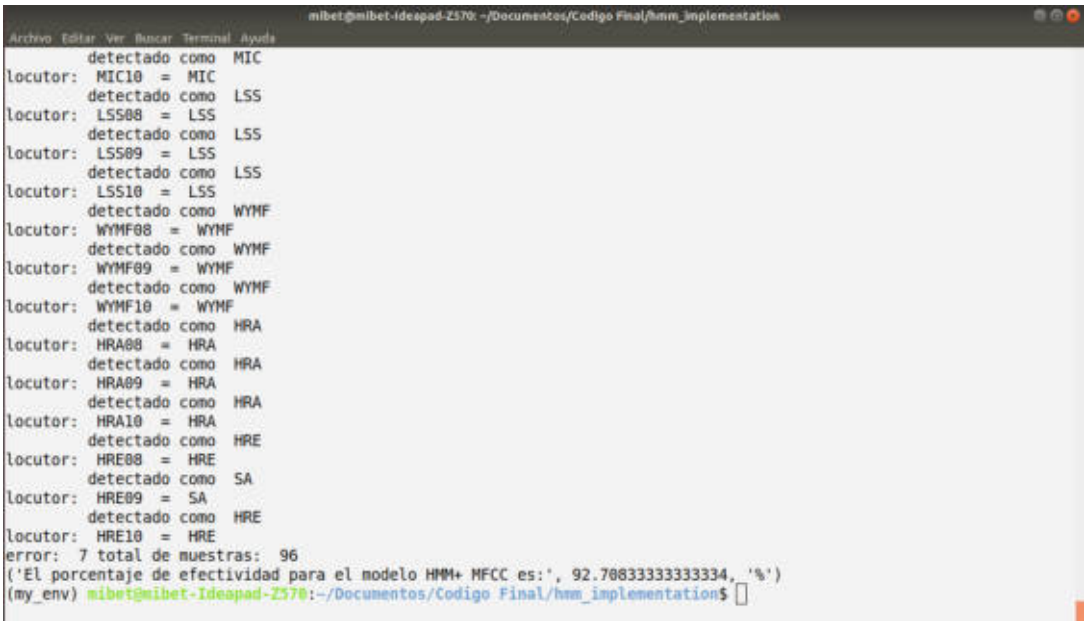

Figura A.3 — Prueba del método HMM y Acurracy (32 docentes)

2. Codificación nativa de GMM

Inicialización de la ejecución de la codificación nativa de GMM. Seguidamente se ejecuta el entrenamiento en los GMM.

Figura A.4 — Entrenamiento del método GMM (32 docentes)

Datos de ejecución del entrenamiento del GMM, con 7 audios por docente.

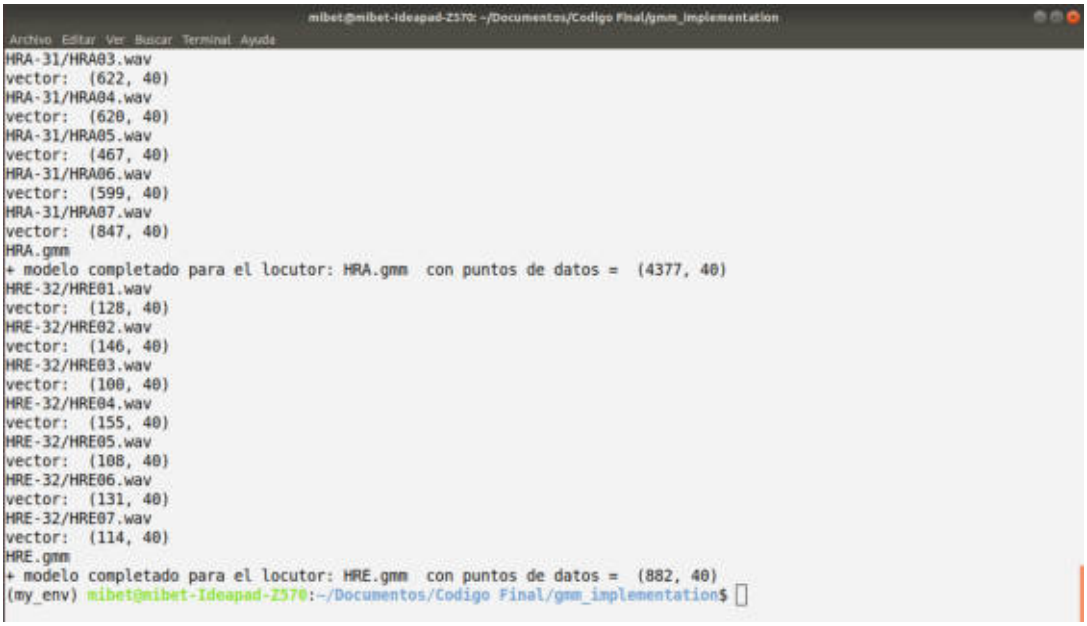

Figura A.5 — Creando modelos para cada docente con GMM.

Ejecución del Test del GMM, cargando los 32 docentes entrenados y la identificación de los 32 docentes.

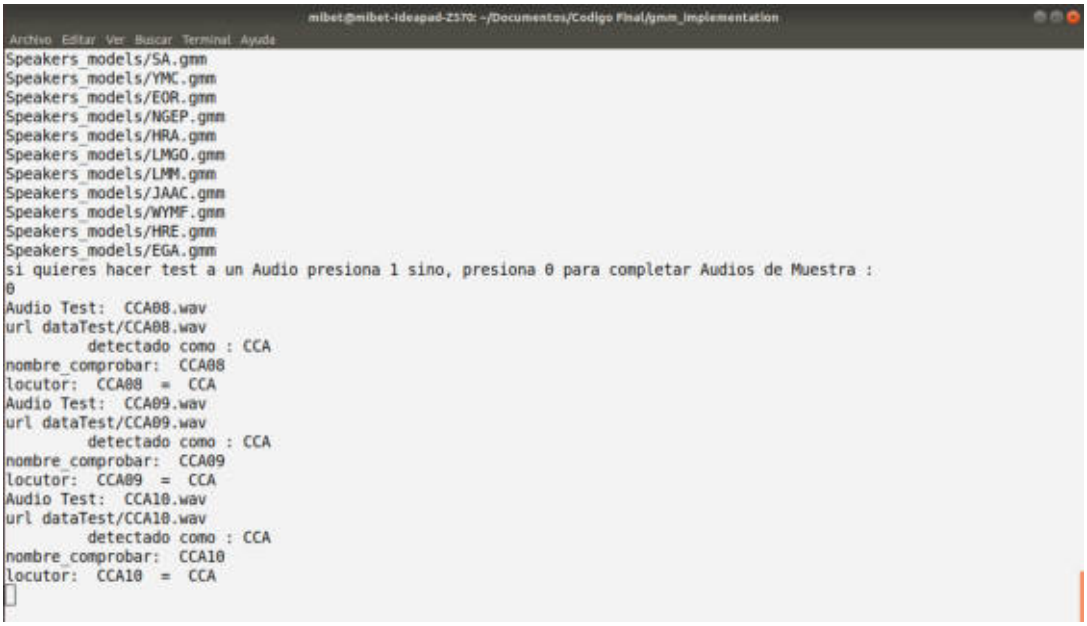

Figura A.6 — Carga de modelos entrenados con el método GMM (32 docentes)

Con Resultados de 2 errores de 96 audios, dando un Accuracy de 97.91.

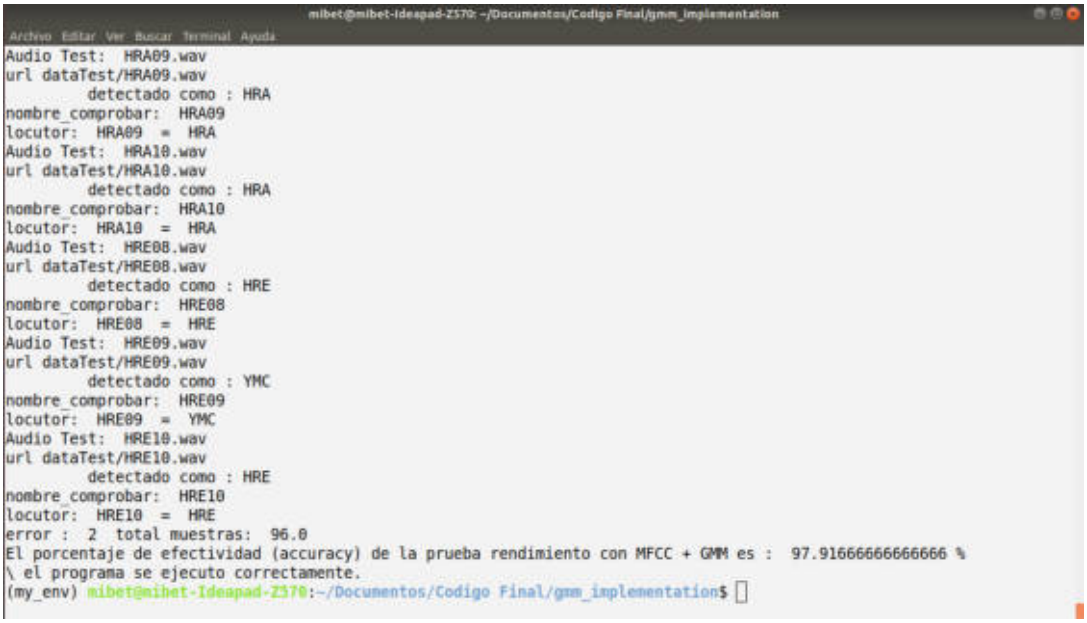

Figura A.7 — Prueba del método GMM y Acurracy (32 docentes)

3. Codificación nativa de SVM

Inicialización de la ejecución de la codificación nativa de SVM. Carga los docentes previamente entrenados y nos muestra opciones a escoger como el "entrenamiento" y "test"

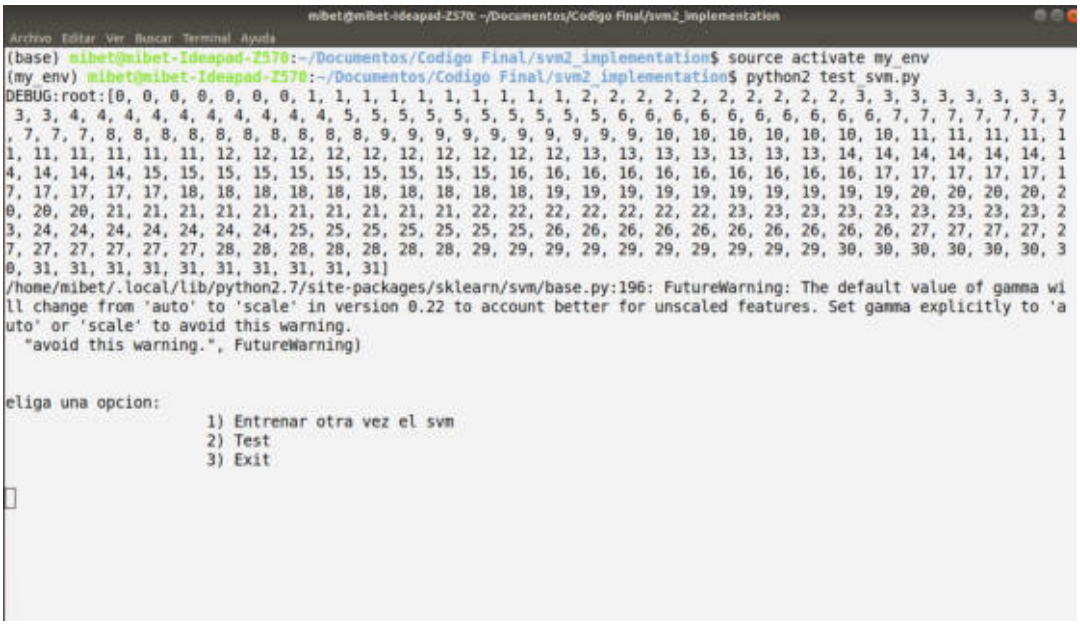

Figura A.8 — Inicio del método SVM

Resultados de SVM, con 27 errores de 96 audios, dando un Accuracy de 71.875%.

Figura A.9 — Prueba del método SVM y Acurracy (32 docentes)

4. Texto de VoxForge

"No sé qué día de agosto del año 1816 llegó a las puertas de la Capitanía General de Granada cierto haraposo y grotesco gitano, de sesenta años de edad, de oficio esquilador y de apellido o sobrenombre Heredia, caballero en flaquísimo y destartalado burro mohíno, cuyos arneses se reducían a una soga atada al pescuezo; y, echado que hubo pie a tierra, dijo con la mayor frescura «que quería ver al capitán general.» Excuso añadir que semejante pretensión excitó sucesivamente la resistencia del centinela, las risas de las ordenanzas y las dudas y vacilaciones de los edecanes antes de llegar a conocimiento del Excelentísimo Sr. D. Eugenio Portocarrero, conde del Montijo, a la sazón capitán general del antiguo reino de Granada…. Pero como aquel prócer era hombre de muy buen humor y tenía muchas noticias de Heredia, célebre por sus chistes, por sus cambalaches y por su amor a lo ajeno…, con permiso del engañado dueño, dio orden de que dejasen pasar al gitano. Penetró éste en el despacho de Su Excelencia, dando dos pasos adelante y uno atrás, que era como andaba en las circunstancias graves, y poniéndose de rodillas exclamó: ¡Viva María Santísima y viva su merced, que es el amo de toitico el mundo! Levántate; déjate de zalamerías, y dime qué se te ofrece

respondió el conde con aparente sequedad. Heredia se puso también serio, y dijo con mucho desparpajo: Pues, señor, vengo a que se me den los mil reales. ¿Qué mil reales? Los ofrecidos hace días, en un bando, al que presente las señas de Parrón. Pues ¡qué! ¿tú lo conocías? No, señor. Entonces…. Pero ya lo conozco. ¡Cómo! Es muy sencillo. Lo he buscado; lo he visto; traigo las señas, y pido mi ganancia. ¿Estás seguro de que lo has visto? exclamó el capitán general con un interés que se sobrepuso a sus dudas. El gitano se echó a reír, y respondió: ¡Es claro! Su merced dirá: este gitano es como todos, y quiere engañarme. ¡No me perdone Dios si miento! Ayer vi a Parrón. Pero ¿sabes tú la importancia de lo que dices? ¿Sabes que hace tres años que se persigue a ese monstruo, a ese bandido sanguinario, que nadie conoce ni ha podido nunca ver? ¿Sabes que todos los días roba, en distintos puntos de estas sierras, a algunos pasajeros y después los asesina, pues dice que los muertos no hablan y que ése es el único medio de que nunca dé con él la Justicia? ¿Sabes, en fin, que ver a Parrón es encontrarse con la muerte? El gitano se volvió a reír."