



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

***IDENTIFICACIÓN DE
GÉNEROS MUSICALES***

TESIS

Que para obtener el título de

INGENIERO EN COMPUTACIÓN

P R E S E N T A

FRINE ANAID SOLANO HERNÁNDEZ

DIRECTOR DE TESIS

DR. IVÁN VLADIMIR MEZA RUIZ



Ciudad Universitaria, Cd. Mx., 2019

Índice

Capítulo I.....	8
Introducción	8
1.1 Antecedentes.....	10
1.1.1 Música	10
Géneros musicales.....	11
1.2 Objetivo.....	13
1.4 Definición del problema	13
1.5 Organización.....	13
Capítulo II.....	15
Trabajo Previo.....	15
2.1 Clasificación de género musical por medio de señales de audio.	15
2.2 Sistema de clasificación de género-musical basado en características espectro-temporales y selección de funciones	19
Capítulo III.....	23
Herramientas a utilizar.....	23
3.1 MFCC.....	23
3.2 Árboles de decisión.....	28
3.2.1 Construcción del árbol de decisión	29
3.3 Random Forest.....	33
3.3.1 Algoritmo de generación de bosque de árboles aleatorios.....	33
3.4 Validación cruzada.....	34
3.5 Máquinas de soporte vectorial.....	36
Capítulo IV	37
Metodología	37
4.1 Base de datos de género.....	37
4.2 Arquitectura del sistema.	38
Capítulo V	41
Experimentos y resultados.....	41
5.1 Experimentos.	41
5.1.1 Experimento con corpus desbalanceado.....	42

Resultado del experimento con corpus desbalanceado	43
5.1.2 Experimento con corpus balanceado	44
Resultado del experimento con corpus balanceado	45
Análisis de métricas entre los dos corpus.....	46
5.1.3 Experimento con corpus externo	46
Resultado del experimento con corpus externo	47
Capítulo VI.....	49
Conclusiones.....	49
Apéndice.....	52
Código utilizado.....	52
Crea_modelo.py	52
1_sonidos_a_csv_ventana_id.py	54
2_archivo_csv_final.py	56
3_evaluacion_v8_id.py	57
Resultados	65
Resultado del experimento con corpus desbalanceado	65
Resultado del experimento con corpus balanceado	65
Resultado del experimento con corpus externo	66
Bibliografía	67
Trabajos citados	67

Índice de tablas

TABLA 1 RESULTADOS EN MATRIZ DE CONFUSIÓN	21
TABLA 2 HERRAMIENTAS UTILIZADAS EN CADA ARTICULO DE TRABAJOS PREVIOS REALIZADOS.....	21
TABLA 3 CONTENIDO DEL CORPUS DE DATOS CON LA DISTRIBUCIÓN DE LOS GÉNEROS MUSICALES	38
TABLA 4 BASE DE DATOS CON GENEROS DESBALANCEADOS.....	42
TABLA 5 RESULTADOS DE LA EVALUACIÓN DEL EXPERIMENTO CON EL CORPUS DESBALANCEADO	43
TABLA 6 MÉTRICAS DE LA EVALUACIÓN DEL EXPERIMENTO CON EL CORPUS DESBALANCEADO.....	43
TABLA 7 CONTENIDO DE CANCIONES DEL EXPERIMENTO CON EL CORPUS BALANCEADO	44
TABLA 8 RESULTADOS DE LA EVALUACIÓN DEL EXPERIMENTO CON EL CORPUS BALANCEADO.	45
TABLA 9 MÉTRICAS DE EVALUACION DEL EXPERIMENTO CON CORPUS BALANCEADO.	45
TABLA 10 COMPARATIVO DE LAS METRICAS DE EVALUACIÓN DE LOS EXPERIMENTOS CON EL CORPUS BALANCEADO Y DESBALANCEADO.	46
TABLA 11 CONTENIDO DE CANCIONES DENTRO DEL CORPUS EXTERNO.....	46
TABLA 12 MATRIZ DE CONFUSIÓN DE LA VALIDACIÓN CRUZADA DEL CORPUS EXTERNO.....	47
TABLA 13 MATRIZ DE CONFUSIÓN DEL EXPERIMENTO CON EL CORPUS EXTERNO.	47
TABLA 14 RESULTADOS DE MÈTRICAS DEL EXPERIMENTO CON EL CORPUS EXTERNO.	48

Índice de figuras

FIGURA 1 DIAGRAMA DE FLUJO DE HISTOGRAMA DE IMPULSOS DEL ARTÍCULO CLASIFICACIÓN DE GÉNEROS MUSICALES DE SEÑALES DE AUDIO.....	18
FIGURA 2 DIAGRAMA DE FLUJO DEL SISTEMA DE CLASIFICACIÓN DE GÉNEROS MUSICALES PROPUESTO.	20
FIGURA 5 COEFICIENTES MFCC.....	24
FIGURA 6 DIAGRAMA DE FLUJO DE LA OBTENCIÓN DE LOS MFCC'S	24
FIGURA 7 REPRESENTACIÓN DE LOS NÚMEROS QUE HACEN REFERENCIA A LA AMPLITUD DE LA SEÑAL, CONTENIDOS DENTRO DEL ARCHIVO .WAV.....	25
FIGURA 8 VENTANA DE HAMMING.	26
FIGURA 9 SUAVIZADO DE LA SEÑAL DE AUDIO CON LA VENTANA DE HAMMING.	26
FIGURA 10 CÁLCULO DEL ESPECTRO DE LA SEÑAL DE LA VENTANA DEL PASO 2	27
FIGURA 11 FRECUENCIAS AUDIBLES AL OIDO HUMANO DE LA SEÑAL A PARTIR DEL FILTRO MEL.....	27
FIGURA 12 COEFICIENTES MFCC'S	28
FIGURA 13 ÁRBOL DE DECISIÓN DE EDADES.....	28
FIGURA 14 ÁRBOL DE CLASIFICACIÓN CON DOS NIVELES DE PROFUNDIDAD.....	29
FIGURA 15 EJEMPLO DE PUNTOS A CLASIFICAR CON BASE A LAS COORDENADAS (X,Y).	30
FIGURA 16 ÁRBOL GENERADO AL REALIZAR EL PARTICIONAMIENTO EN LA COORDENADA Y.	31
FIGURA 17 ÁRBOL GENERADO AL REALIZAR EL PARTICIONAMIENTO EN LA COORDENADA X.	31
FIGURA 18 ÁRBOL FINAL CON LAS PROBABILIDADES DE CADA NODO A PERTENECER A UNA CLASE, CON BASE A LA RAMA QUE SIGUIÓ PARA DESCENDER.	32
FIGURA 19 PARTICIONAMIENTO DEL CORPUS EN DIEZ SECCIONES.	35
FIGURA 20 PROCESO DE ITERACIÓN DE LAS ESTRATIFICACIONES DEL CORPUS.	35
FIGURA 21 SISTEMA DE ENTRENAMIENTO	39
FIGURA 22 SISTEMA IDENTIFICADOR DE ARCHIVOS.....	39
FIGURA 23 SISTEMA IDENTIFICADOR EN VIVO	40
FIGURA 24 RESULTADOS DE LA EVALUACIÓN DEL EXPERIMENTO CON EL CORPUS DESBALANCEADO	65
FIGURA 25 RESULTADOS DE LA EVALUACIÓN DEL EXPERIMENTO CON EL CORPUS BALANCEADO.	65
FIGURA 26 MATRIZ DE CONFUSIÓN DE LA VALIDACIÓN CRUZADA DEL CORPUS EXTERNO.....	66

Capítulo I

Introducción

En este trabajo se presenta un detector automático de género musical usando la técnica de máquinas de soporte vectorial (SVM, por sus siglas en inglés “*support vector machine*”). El objetivo del sistema es “escuchar” una canción en formato digital y determinar el género de éste. En esta sección se describe las motivaciones de desarrollar dicho sistema.

Motivación

El principal motivo de desarrollo de esta tesis es el de dotar a un sistema computacional con las capacidades de entender la música, en particular la tesis se limita al género musical. Desde que el humano comenzó a poblar la tierra, la música ha sido uno de los principales medios de comunicación y unificación de diferentes ideologías. La música a lo largo del tiempo ha ayudado a los seres humanos a comunicarse, concentrarse, relajarse y relacionarse con otros seres humanos.

El impacto que la música ha generado en el aspecto sociológico, ha sido ampliamente estudiado y existen un sinnúmero de investigaciones realizadas por diversas áreas de la ciencia como lo son: las matemáticas (Alain Danielou, Introduction to the Study of Musical Scales, Mushiram Manoharlal 1999, Chapter 1 *passim.*), medicina (The Parkinson disease quality

of life assessment (PDQ-39), Peto, Jenkinson & Fitzpatrick, 1998), psicología (*The Oxford Handbook of Music Psychology*. Oxford: Oxford University Press. p. 539., 2009), sociología (DeNora, Tia. 2000. *Music in everyday life*. Cambridge, UK: Cambridge Univ. Press.) y más. En los últimos años ha habido un creciente interés por adaptar sistema de computación e interacción humana a ser capaces de manipular la música (Computational Counterpoint Worlds. *Mathematical theory, software and experiments*, Agustin Aquino, 2015).

Una vez definidos los aspectos con los que se trabajará en este escrito, este trabajo se encuentra dentro de una de las áreas más interesantes de la inteligencia artificial: el aprendizaje automático. El aprendizaje automático o "Machine Learning" en inglés, es una subarea de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas para que los computadores logren implementar técnicas que modelen mediante modelos matemáticos un comportamiento. De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de una información suministrada en forma de ejemplos, en nuestro caso de un segmento de una canción a un género de música predefinido.

El aprendizaje automático tiene una amplia gama de usos, incluyendo grandes tecnologías que se emplean en el día a día de nuestras vidas, como lo son: identificación de canciones filtrado de spam, identificación de rostros, diagnósticos médicos, análisis de mercado, reconocimiento de patrones de lenguaje, juegos, robótica, predicción climatológica, entre otros.

La finalidad de este trabajo es la de desarrollar un sistema informático que logre reconocer y diferenciar exitosamente los géneros musicales seleccionados: Cumbia, Salsa, Rock y Pop. Esto tendría en el futuro lograr que el sistema pueda ser usado para generar vínculos entre los usuarios con afinidad al mismo género musical, como supuesto en donde el usuario pueda entrar a un lugar y sólo con el entrar y escuchar lo que está oyendo, el sistema pueda reconocer que género es y en un futuro se pueda implementar para sistemas automatizados de música.

En la siguiente sección se detallarán los puntos que buscaremos abarcar dentro de este trabajo, para poder desarrollar el sistema el cuál permita identificar los cuatro géneros musicales anteriormente mencionados.

1.1 Antecedentes

Desde la antigüedad la música ha ocupado un lugar importante. Se cree que el primer instrumento musical utilizado por el ser humano haya sido su propio cuerpo, ya que este permite realizar sonidos por medio de sus extremidades, de los sonidos vocales que este producía o del uso de instrumentos creados por el mismo hombre.

Desde sus orígenes la música se ha usado como un medio de comunicación. Puede ser que el hombre de la antigüedad haya descubierto el poder que tiene la música sobre los estados de ánimo del individuo, asociándolo a los diferentes usos que le podrían dar en ese momento. (Lobo, 2003)

Partiendo de este punto la música produce un efecto de relajación en los sentidos, que pueden ser utilizados dependiendo los intereses del individuo, algunos ejemplos podrían las marchas militares, las cuales producían un sentido colectivo de unión y fuerza. En las antiguas culturas la música era utilizada para fines curativos a diferencia de las marchas militares que son utilizadas para la guerra.

Lo que nos permite apreciar el poder que tiene la música sobre el individuo para poder alcanzar diferentes objetivos así como influir en el estado de ánimo ya sea de manera individual o colectiva.

1.1.1 Música

“La historia de la música se remonta hasta los inicios del ser humano, ya que empezó con los sonidos que emanaban, así como en los ruidos que hacían al hacer sus actividades cotidianas como sería la caza, entre otros.” - *José María Morales de Ávala*

Cada estilo tiene su momento en el cual debe ser escuchado, para lo cual tiene varios elementos importantes:

- Un elemento fundamental es el ritmo.
- Cada música tiene su propio volumen.
- Los modos o tonalidades en los que se desarrollan las canciones.
- El elemento principal es la letra. (Morales de Álava José María, 1999)

Poco a poco la música se fue asociando a la danza pues se decía que tenía un poder espiritual y mágico. Utilizando los instrumentos de viento y percusión. La danza y la música van unidas ya que estaban relacionadas con el movimiento y con la forma de invocar a la naturaleza. Con la música, el hombre se aísla por un momento de sus preocupaciones externas y se hace uno con ella, esto sirve además para unir a los hombres. No porque la música sea condición necesaria para que la sociedad exista, sino porque facilita un sentimiento común más aun, exige un esfuerzo de concordancia. (Guervós, 2008)

En el trabajo de *Nicolas Scariengella* vemos las formas en que podía realizar un análisis de la música para poder buscar identificar el género musical de una canción. Este trabajo nos presenta propuestas para analizar la música mediante la armonía, el ritmo y el timbre de la canción, de los cuales tenemos que la armonía no es conveniente ya que cada canción tiene una diferente. Mientras que por otro lado el ritmo no es lo mejor ya que dentro de una misma canción se presentan diferentes ritmos, finalmente tenemos la propuesta de analizar con base al timbre el cual incluye todo, desde la voz hasta los instrumentos musicales (Nicolas Scaringella, 2006).

Géneros musicales

Se emplea en música para clasificar las obras musicales, como esta clasificación se puede hacer de distinta forma dependiendo de los criterios que se utilicen para realizarla, (según los medios sonoros, la función, los contenidos, etc.), se habla de géneros musicales.

Se conoce como género musical a la categoría que reúne composiciones musicales que comparten diferentes criterios, instrumentos, contexto social.

A continuación, se describe cada uno de los géneros utilizados en este trabajo, lo cuales fueron seleccionados al ser de los más escuchados dentro de la población de la CDMX:

Salsa

El género musical salsa es la combinación del son cubano con síntesis de música caribeña y jazz así como otros ritmos estadounidenses. Es considerado un géneroailable el cual se consolidada como un éxito comercial por músicos de origen puertorriqueño en Nueva York en la década de 1960, siendo hasta el momento uno de los géneros de mayor aceptación (SuperRadio, 2013)

Pop

Su historia remota desde el siglo XX, bajo la influencia del Rock y el Doo-wop. Las canciones pop se basan en la repetición de melodías en torno a una construcción clásica de estrofa, puente y estribillo.

Rock

Este género musical nace en pleno siglo XX. El mismo deriva del “Rock and Roll” además de otros estilos de músicas, entre los que se puede citar al country, el góspel, el jazz y el blues, principalmente del último nombrado ya que es un estilo muy rápido.

Cumbia

Este género musical que representa a Colombia en el mundo es un fiel testimonio de la riqueza rítmica y cultural del país. Tiene su origen de tres vertientes culturales,

principalmente indígena y negra africana y, en menor medida, blanca (española) (Denizeau, 2005).

Se agregó otro género denominado ruido el cual engloba otros géneros, así como el ruido ambiental.

1.2 Objetivo

Desarrollar un sistema el cual pueda detectar e identificar diferentes géneros musicales. En este trabajo se delimito a solo 4 géneros los cuales son cumbia, pop, rock y salsa, de los cuales se habló un poco de sus orígenes anteriormente.

1.4 Definición del problema

Dada una señal de audio correspondiente a música determinar el género del cual corresponde esta señal de audio, teniendo como problema que los cuatro géneros tienen cierta relación entre ellos lo cual hace un poco complicado poder diferenciarlos

1.5 Organización

A continuación, se presenta la organización de este trabajo.

En el Capítulo II: Se muestra cómo este tema no es nuevo en la investigación y la manera de cómo han trabajado algunos investigadores, como se muestra en los trabajos de *Tzanetakis* y *Lim*.

En el Capítulo III: Se muestra cuáles son las herramientas que se utilizarán para poder llevar a cabo cada uno de los procesos de la tesis, en este trabajo se muestra MFCC'S y máquina de aprendizaje utilizando Random Forest.

En el Capítulo IV: Se podrá observar la metodología que se usa para posteriormente realizar las pruebas. Se define la base de datos y la arquitectura que se utilizará.

En el Capítulo V: Se observa cada uno de los experimentos que se realizaron en las pruebas para poder así llegar a diferentes resultados, dependiendo el tipo de prueba.

En el Capítulo VI: Finalmente en este capítulo se llega a las conclusiones dependiendo de cuales fueron los resultados obtenidos en las pruebas realizadas.

Capítulo II

Trabajo Previo

El estudio de detección de género tiene una tradición en la computación, a continuación se presentan unos trabajos que se han enfocado a la tarea de la identificación de géneros musicales haciendo uso de una computadora, con los que utilizando un corpus, técnicas de extracción de características y máquinas de aprendizaje, se busca realizar la identificación de géneros musicales. Se puede ver en el trabajo de *Tzanetakis* el cual menciona que el humano crea música para describir al universo, mediante la relación de los instrumentos con el ritmo (Tzanetakis, 2002). Mientras que *Lim* busca examinar cómo identificar los géneros musicales mediante la utilización de características de espectro temporal y selección de características (Lim, 2012).

A continuación, revisaremos con mayor detalle lo anterior internándonos un poco en los trabajos previamente desarrollados.

2.1 Clasificación de género musical por medio de señales de audio.

Tzanetakis menciona que la música para el humano es creada y clasificada para describir el universo. (Tzanetakis, 2002) Una manera particular de clasificar la música es mediante la relación de la instrumentación y la estructura del ritmo.

Al querer poner en una jerarquía de los géneros musicales se presenta un problema al momento de la clasificación, debido a que existen tres conjuntos que ayudan a representar

las características del audio; las cuales son la textura del timbre, el contenido del ritmo y el contenido del tono.

Los autores trabajan con la idea de que todo tipo de sistema de base de análisis de audio es la extracción de vectores característicos. La extracción de características es el proceso del cálculo de su representación numérica compacta que puede ser utilizado para caracterizar un segmento de audio.

Para lo cual los autores manejan tres características principales:

a. Características de la textura del timbre

El timbre es la cualidad del sonido que permite distinguir los diferentes sonidos, debido a la estructura misma del sonido. (ALICIA, 2007)

Para representar la textura del timbre se basada en un estándar de características propuesto para música, discurso y reconocimiento de voz. Calcular dichas características está basado en el pequeño tiempo de la trasformada de Fourier de Tiempo Corto (STFT) y el cálculo de cada fragmento de sonido.

- Centroide espectral: es definido como el centro de gravedad de la magnitud del espectro del STFT.
- Atenuación espectral: está definido como la frecuencia R_i con el 85% de la distribución de magnitudes es concentrada. La atenuación es otra forma de medir la forma espectral.
- Flujo espectral: es definido como el cuadrado diferencial entre las magnitudes de normalización de la distribución espectral sucesiva.
- Dominio del tiempo por cruzamiento cero:

$$Z_t = \frac{1}{2} \sum_{n=1}^N |\text{sing}(x[n]) - \text{sing}(x[n-1])|$$

Ecuación 1 Dominio del tiempo por cruzamiento cero

Donde la función *sign* es 1 para los argumentos positivos y 0 para argumentos negativos, y $x[n]$ es el dominio de tiempo por marco de t .

- Coeficientes Espectrales de Frecuencia Mel: (MFCC) Típicamente son usados 13 coeficientes para la representación del discurso, los 5 primeros coeficientes provienen de la mejor clasificación de géneros.
- Análisis y textura de ventanas: en un pequeño tiempo de análisis de audio. Este segmento es llamado análisis de ventanas y tiene que ser lo suficiente pequeño, así la frecuencia característica del espectro magnitud es relativamente estable.
- Característica baja-energía: está basada en la textura y análisis de ventanas. Es definido como el porcentaje de análisis de ventanas al tener menor energía RMS que la media RMS de cruzamiento de energía de textura de ventanas.

b. Características del contenido del ritmo

El vector de características de la textura de timbre consiste en las características siguientes: medias y varianzas del centroide espectral, atenuación flujo, cruzamiento cero sobre la textura de la ventana, baja energía y medias y varianzas de los primeros cinco MFCC coeficientes sobre la textura de la ventana resultando en un vector de 19 características.

c. Características del contenido del ritmo

Sistemas de detección de impulso automático proporciona un corrimiento estimado del impulso inicial y un estimado de su fuerza.

El cálculo de características para representación de la estructura del ritmo de música es basado en la transformada de wavelet (WT) es una técnica de analizar señales que es desarrollado como una alternativa del STFT.

Diagrama de flujo de histograma de impulsos

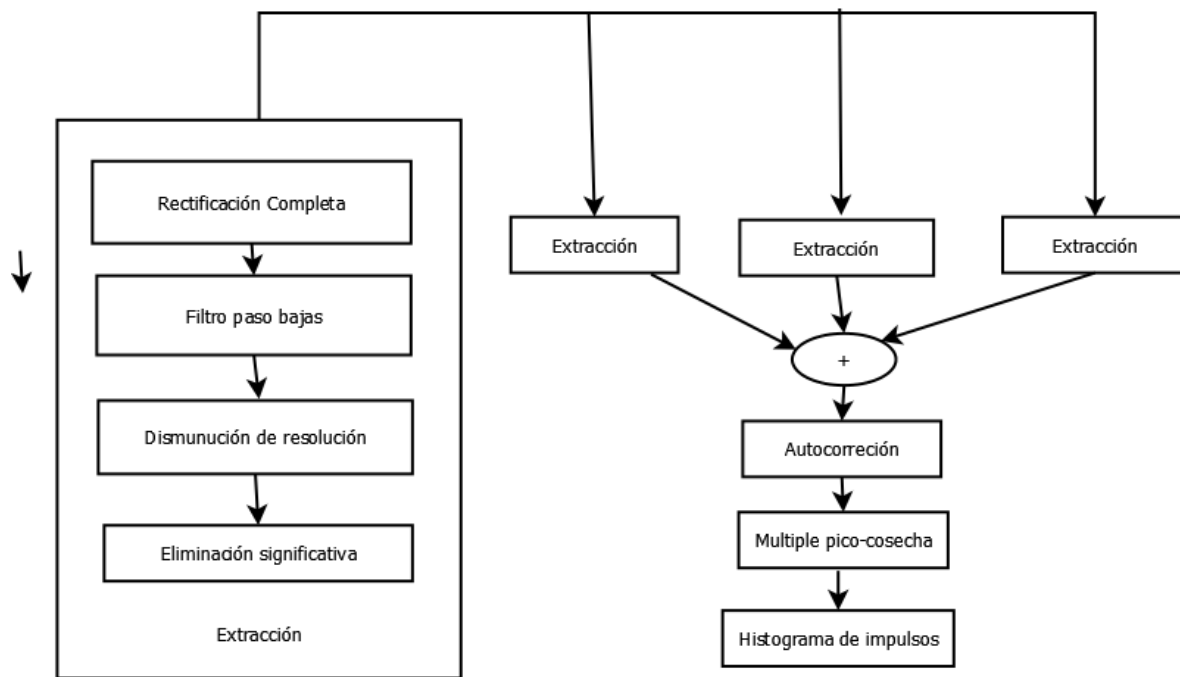


Figura 1 Diagrama de flujo de histograma de impulsos del artículo Clasificación de Géneros Musicales de Señales de Audio.

2.2 Sistema de clasificación de género-musical basado en características espectro-temporales y selección de funciones

Lim busca realizar la identificación de género musicales mediante la utilización de características de espectro temporal y selección de características. (Lim, 2012)

Existen tres niveles de característica para la clasificación de géneros musicales, dentro de los cuales el nivel más alto proporciona las etiquetas para el género, sentimiento y artista. Por otro lado el nivel bajo se encarga de obtener características las cuales son propias del sonido como ritmo y armonía. Las características de nivel medio se encargan de representar las propiedades de la música mediante el tono, el ritmo y la armonía.

Lim investiga cómo realizar la clasificación basada en el uso del clasificador SVM (Support Vector Machine) (Lim, 2012) Indaga realizar la mejora a la clasificación obtenida previamente por otros trabajos en términos de exactitud y complejidad computación, para ello utilizan algoritmos para reducción de dimensiones

El sistema propuesto consiste en dos fases:

- Fase de entrenamiento: esta fase consiste en la extracción de las características del timbre de los archivo de prueba, a estas características se le extrae el espectro temporal, después de eligen las características a analizar, se normalizan y se obtiene el modelo generado.
- Fase de clasificación: en esta fase a los archivos de prueba se le extraen las características del timbre, para que se obtenga el espectro temporal de estas, se seleccionen las mismas características que en la etapa de entrenamiento, se normalice y finalmente se le pase al clasificador para determinar la etiqueta del género de cada archivo de prueba.

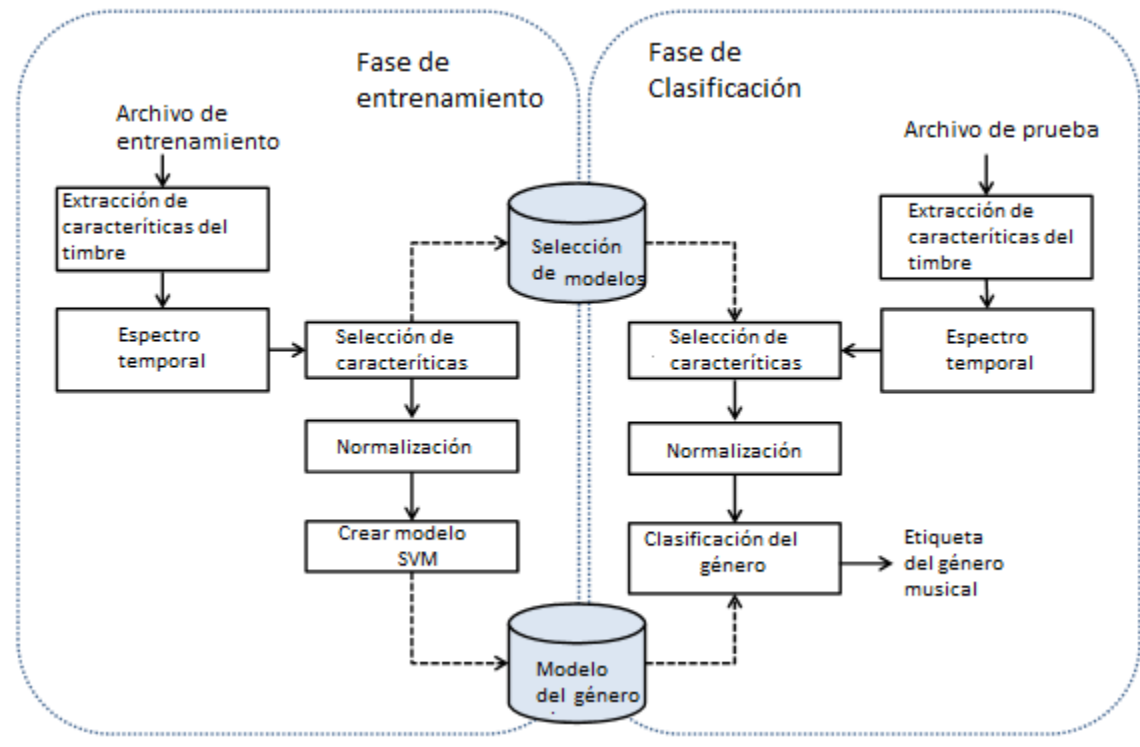


Figura 2 Diagrama de flujo del sistema de clasificación de géneros musicales propuesto.

En la extracción de características para cada uno de los archivos se divide en tres secciones:

- Características basadas en el timbre
 - MFCC, para modelar la forma en que el humano puede escuchar.
 - Banco de filtro correlacionados el cual permite la utilización de un filtro pasa-altas
 - Contraste espectral en octava base, permitiendo modelar la señal de la música
- Función del espectro temporal:
- Selección de funciones y algoritmos de reducción de dimensiones, esto permitió hasta donde menciona el autor que incremento el rendimiento del reconocimiento y disminuyó la complejidad computacional.

	Música clásica	Electrónica	Jazz/ Blues	Metal/ Punk	Rock/ Pop	Mundial
Música clásica	313	1	0	0	0	6
Electrónica	1	105	0	0	4	4
Jazz/Blues	0	0	23	0	2	1
Metal/Punk	0	1	0	37	6	1
Rock/Pop	0	9	1	6	81	5
Mundial	6	15	0	0	5	96

Tabla 1 Resultados en matriz de confusión

Artículo	Artículo 1	Artículo 2
Características del sonido	MFCC's. Contenido del ritmo.	MFCC's.
Máquina de aprendizaje	No se menciona	SVM
Numero de Géneros utilizados	No se menciona	6 géneros musicales

Tabla 2 Herramientas utilizadas en cada artículo de trabajos previos realizados.

Capítulo III

Herramientas a utilizar

Se darán a conocer las herramientas que se utilizarán para la extracción de características del audio mediante MFCC's, y de esta manera tener la información con la que se trabajará siendo como resultado las estadísticas de los MFCC's extraídos, y la máquina de aprendizaje automático utilizado que en este caso es Random Forest. La finalidad de este capítulo es realizar la explicación del funcionamiento de los métodos utilizados.

3.1 MFCC.

Los Mel Frequency Cepstral Coefficients, son los coeficientes empleados para la representación del audio basados en la percepción humana. Estos coeficientes sirven para obtener características representativas, de muestras de audio para su identificación en el área de reconocimiento de voz.

Los parámetros MFCC son un tipo particular de coeficientes cepstrales derivados de la aplicación del Cepstrum sobre una ventana de tiempo de la señal de voz.

Analizando el Cepstrum desde un punto de vista matemático, podemos decir que se trata de un operador que transforma una convolución en el tiempo en una suma en el dominio espectral.

En general, el Cepstrum se define como la transformada inversa de Fourier del logaritmo del espectro de la señal de voz.

$$\text{Cepstrum}(s[n]) = \hat{s}[n] = F^{-1}[\log(|F[s[n]]|)] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(|S(e^{j\omega})|) d\omega,$$

Ecuación 2 Cepstrum

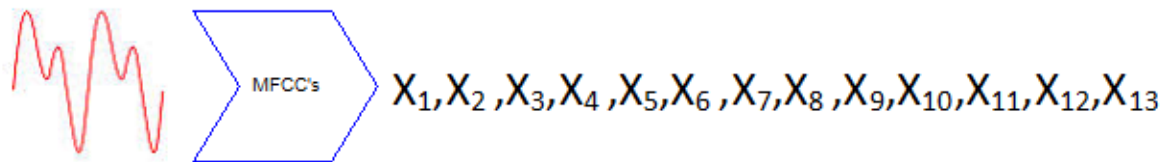


Figura 3 Coeficientes MFCC

¿Cómo obtener los MFCC?

La obtención de MFCC's desde un archivo de audio se obtiene de la siguiente manera:

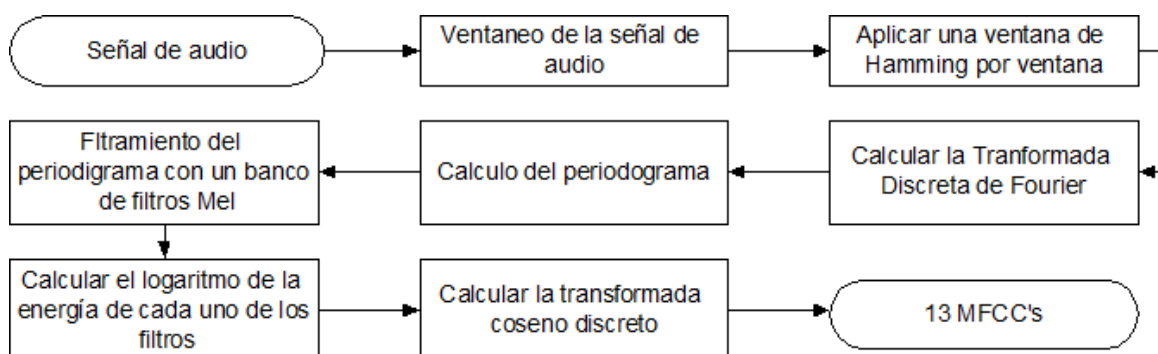


Figura 4 Diagrama de flujo de la obtención de los Mfcc's

En la figura anterior se puede observar el proceso de obtención de MFCC's, donde se tiene por entrada la señal de audio y salida los 13 valores de MFCC's.

El proceso de la obtención de MFCC's se realiza en los siguientes pasos:

1. Ventaneo de la señal de audio
2. Aplicar ventana de Hamming por cada ventana
3. Calcular la transformada discreta de Fourier
4. Cálculo del periodograma
5. Filtramiento del periodograma con un banco de filtros MEL
6. Cálculo del logaritmo
7. Cálculo de la transformada coseno discreto

Paso 1. En este paso se tiene como entrada la señal de audio, la cual se encuentra almacenada dentro de un archivo de audio, en formato WAV. El archivo de audio contiene un vector de números que contiene los valores de amplitud que tiene la señal de audio en determinado tiempo. El ventaneo de la señal se realiza para poder analizar las características de esta pequeña señal dentro de la ventana, ya que si se realiza directamente sobre el audio daríamos por hecho que todos los audios son la misma magnitud, y a la vez no se extraerían las características importantes de la señal.

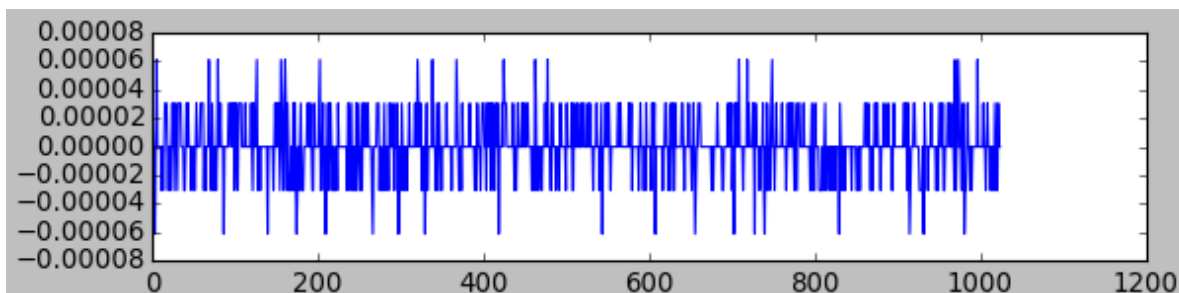


Figura 5 Representación de los números que hacen referencia a la amplitud de la señal, contenidos dentro del archivo .WAV.

Al realizar el ventaneo de la señal se debe de tomar en cuenta que debe ser un tamaño fijo de ventana, y debe de presentar un salto para que la ventana recorra toda la señal de audio.

Por lo que de este paso se obtendrá un conjunto de ventanas que contienen la información de toda la señal de audio, es decir la información del audio a través del tiempo.

Paso 2. Para cada una de las ventanas obtenidas en el paso 1, se le aplicará una ventana de Hamming la cual sea del mismo tamaño que el de la ventana utilizada en el paso anterior.

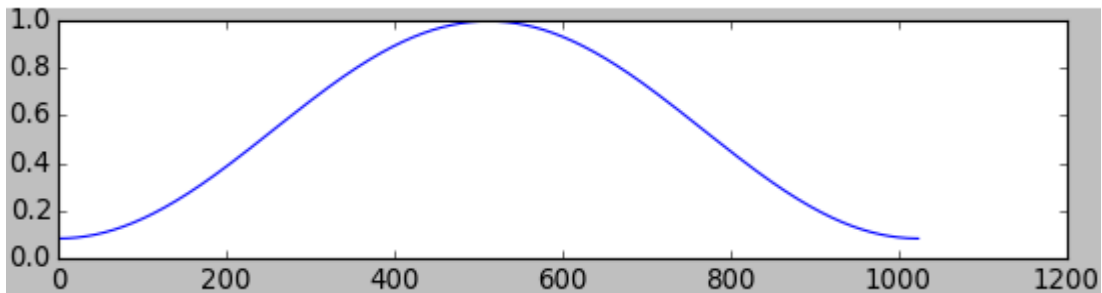


Figura 6 Ventana de Hamming.

El motivo de la aplicación ventana de Hamming es para que se pueda representar esa señal como si fuera una señal cíclica, por lo que se busca que al inicio y final de la ventana los valores de la amplitud de la señal sean similares. Por lo que la salida de este paso es la señal de audio suavizada, en cada una de las ventanas de la señal de audio.

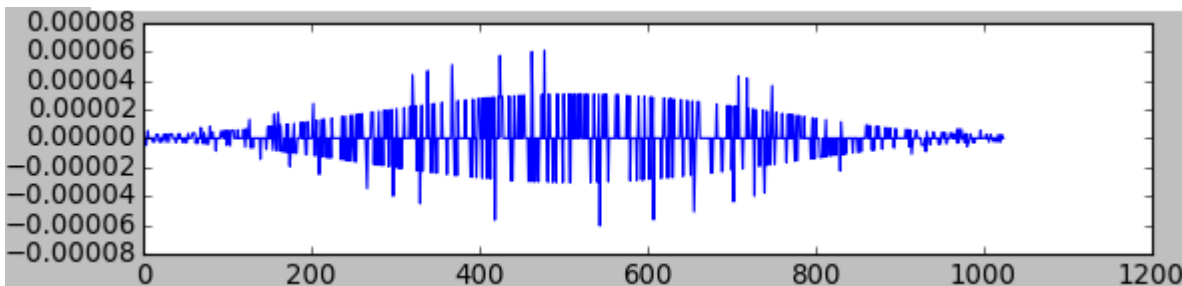


Figura 7 Suavizado de la señal de audio con la ventana de Hamming.

Paso 3. Extracción de la información espectral de cada una de las ventanas de la señal de audio, donde se busca encontrar cuanta energía es la que contiene la señal en diferentes bandas de frecuencia, por lo que se utiliza la Transformada Discreta de Fourier.

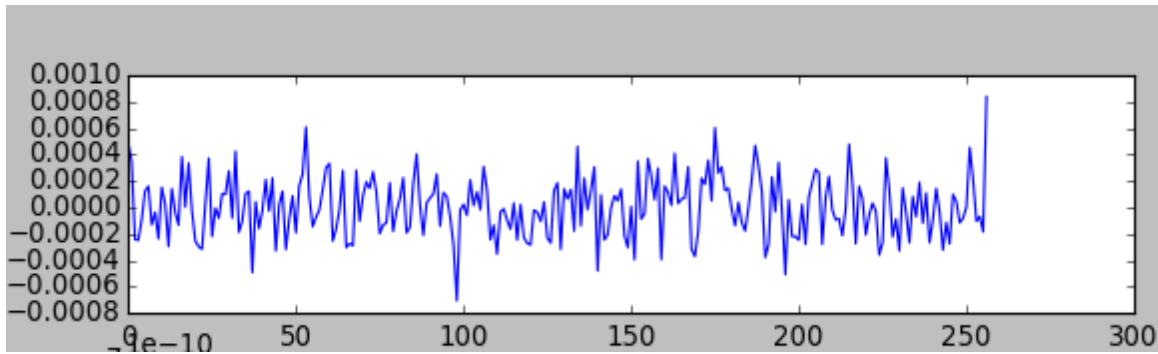


Figura 8 Cálculo del espectro de la señal de la ventana del paso 2

Paso 4. Obtención de la energía de cada una de las bandas, obteniendo la magnitud del vector de la transformada discreta de Fourier para poder eliminar la parte imaginaria de la componente de cada banda, teniendo así el periodograma.

Paso 5. Una vez que se obtuvo la energía de las bandas del paso 4, filtraremos la señal con un banco de filtros MEL, con la finalidad de que se reduzcan las frecuencias existentes y que se tengan aquellas frecuencias en las que el oído humano detecta las variaciones.

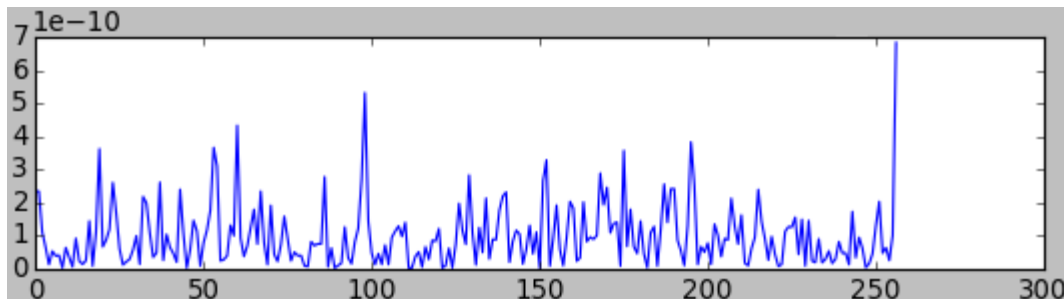


Figura 9 Frecuencias Audibles al oído humano de la señal a partir del filtro MEL.

Paso 6. A cada una de las frecuencias del paso 5, se le calcula el logaritmo.

Paso 7. Calcular la Transformada Discreta Coseno de los logaritmos del espectro (salida del paso 6), para que de esta forma calculemos el espectro del logaritmo del espectro, obteniendo el inverso de la componente espectral (cepstral), para que de esta manera obtengamos los 13 coeficientes MFCC's; 12 son valores cepstrales y 1 coeficiente de energía.

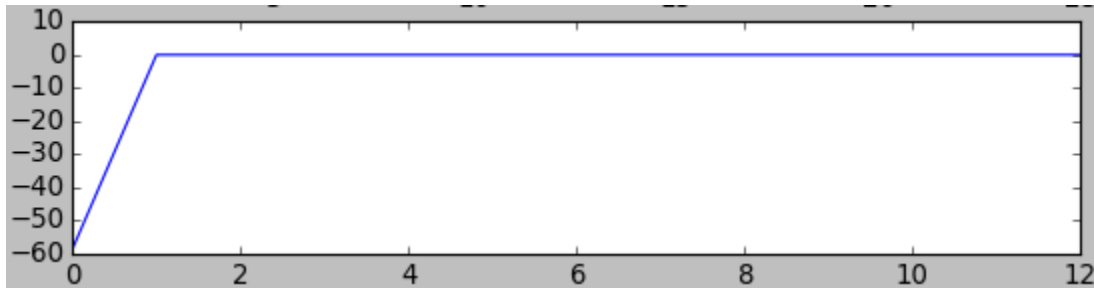


Figura 10 Coeficientes MFCC's

3.2 Árboles de decisión.

Un árbol de decisión es la base para la técnica de aprendizaje automático denominada Árboles Aleatorios (Random Forest en inglés). Por lo que a continuación se presentará su funcionamiento.

Un árbol de decisión es una estructura compuesta por nodos y ramas, los nodos nos representaran las características posibles por analizar mientras que las ramas son las alternativas de los caminos a tomar con base al nodo en que se encuentre, un ejemplo claro es el siguiente:

Deseamos agrupar a la gente de acuerdo a su edad, por lo que la edad que separa a los dos grupos es de menores de 18 años y los demás, por lo que solo tendríamos dos grupos.

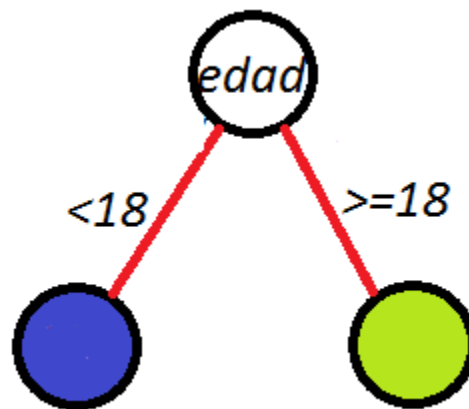


Figura 11Árbol de decisión de edades

En la figura anterior podemos observar que el árbol de decisión cuenta con 3 nodos, de los cuales 1 es el nodo raíz ya que de este parte el árbol y cuenta con 2 nodos hoja, los cuales representan la clase en la que estamos clasificando en el ejemplo.

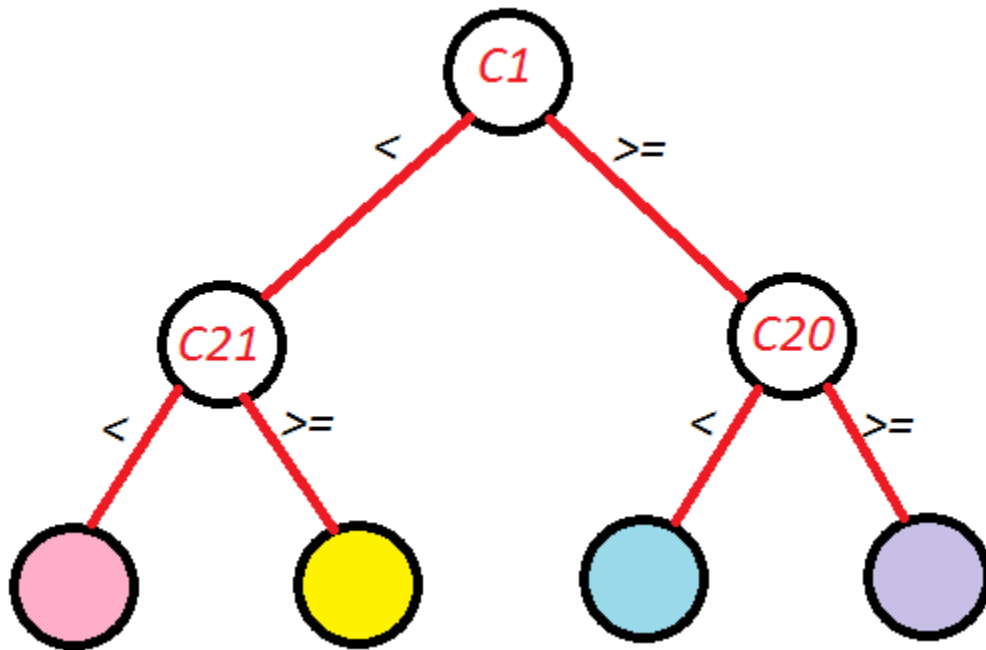


Figura 12 Árbol de clasificación con dos niveles de profundidad

Un árbol de decisión se basa en el uso de características para clasificar a un elemento dentro de una clase. Para que podamos clasificar a un elemento primero debemos de comparar la característica que está en la raíz para que con base a eso se conozca a que nodo siguiente avanzará.

3.2.1 Construcción del árbol de decisión

Un árbol de decisión se basa en la creación de una estructura que permita separar elementos de diferentes clases con base a la comparación de características propias de los elementos, para crear el árbol realiza el siguiente procedimiento:

1) Analizar la entropía cada una de las características de los elementos que permitirá elegir la característica con menor entropía, la cual permita la mejor discriminación de los elementos con base a esa característica, por lo que esta característica será seleccionada como la raíz del árbol.

2) Con base a la característica del nodo raíz se analizará una nueva característica por cada nodo hijo del nodo padre, pero la característica que se elija como la mejor, con menor entropía, no puede debió utilizarse en alguno nodo antecesor del nodo en turno. Si la entropía en ese nodo es la óptima no se elegirá ninguna característica y será un nodo hoja, en caso de que la entropía sea grande se generaran nuevos hijos.

3) El paso [2] se repetirá por cada nodo hijo, hasta que se llegue a la menor entropía en los nodos hijos.

Dentro de cada uno de los nodos del árbol generado se contarán con estadísticas del porcentaje de la clase de cada clase con base a la evaluación realizada por las características.

Para tener más claro cómo se realiza el árbol revisemos el siguiente ejemplo:

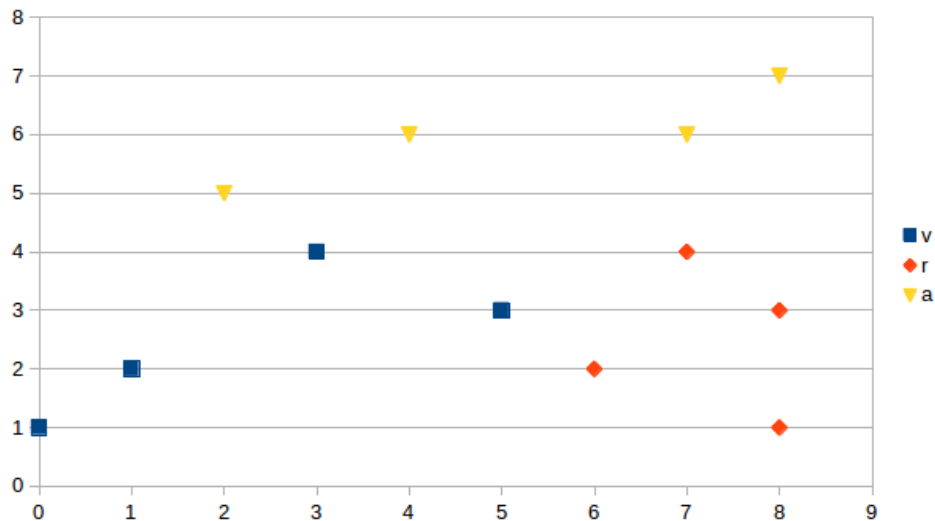


Figura 13 Ejemplo de puntos a clasificar con base a las coordenadas (X,Y).

Elementos a clasificar con base a las coordenadas.

Se creará un árbol de decisión para que con base a las coordenadas de los puntos determinemos las clases v, r ó a.

En este ejemplo sólo existen dos características a evaluar las coordenadas X y Y. Buscando el valor que permita diferir entre los grupos con base a las características tenemos los siguientes:

- $Y \geq 5$ y $Y < 5$

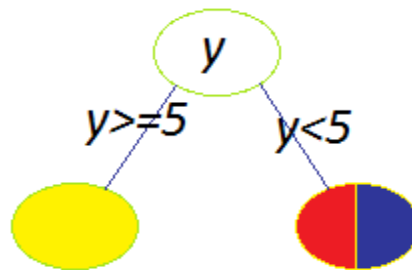


Figura 14 Árbol generado al realizar el particionamiento en la coordenada Y.

- $X \leq 5$ y $X > 5$

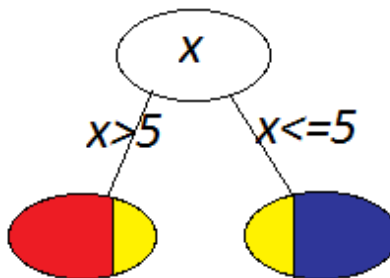


Figura 15 Árbol generado al realizar el particionamiento en la coordenada X.

En ambos casos vemos que la característica separa a los tres grupos pero vemos que la entropía es mero en el caso de la utilización de la característica Y., por lo que es el nodo raíz dicha característica, teniendo que en el nodo hijo izquierdo que el 100% de la población es del grupo a, mientras que en el nodo derecho 50% de la población es v y el otro 50% es r, pudiendo decir que la probabilidad de que un punto que tenga una $Y < 5$ y sea r o v, es de 0.5, mientras que si $Y \geq 5$ será a.

Procederemos a analizar el nodo izquierdo con la característica X, pero como ya tiene la menor entropía es un nodo hoja. El nodo derecho cuenta con una entropía alta por lo que se utilizara la característica X, para disminuirla encontrando que el valor para discriminar estos grupos es el valor de 5, quedando de la siguiente manera el árbol.

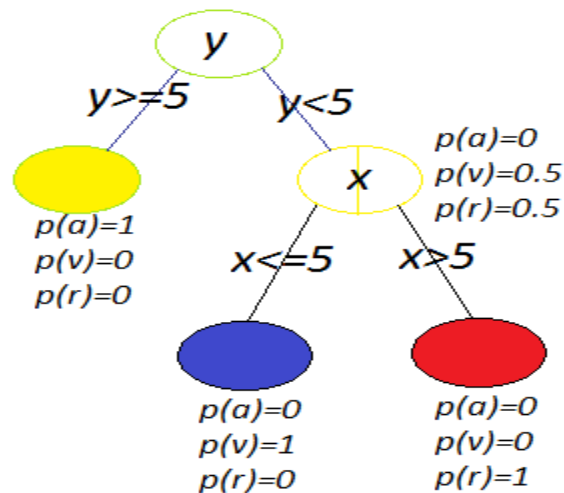


Figura 16 Árbol final con las probabilidades de cada nodo a pertenecer a una clase, con base a la rama que siguió para descender.

Con las probabilidades de la siguiente manera:

$$\text{Si } Y \geq 5, P(a) = 1$$

$$\text{Si } Y < 5, P(v) = 0.5$$

$$\text{Si } Y < 5, P(r) = 0.5$$

$$\text{Si } Y < 5, X \leq 5, P(v) = 1$$

$$\text{Si } Y < 5, X > 5, P(r) = 1$$

3.3 Random Forest

Es un método de aprendizaje automático supervisado, este método utiliza la técnica de “divide y vencerás”, dado que se encarga de generar un conjunto de clasificadores débiles que en conjunto son un clasificador fuerte. Se tomará en cuenta que cada clasificador débil es un árbol de decisión que en conjunto con más árboles de decisión, generarán un bosque, y este bosque será aleatorio.

Random Forest es utilizado para ayudar a la tarea de clasificación en volúmenes altos de información, tomando en cuenta al modelo propuesto por Brieman, integra las siguientes herramientas:

- Árboles de decisión
- Bagging: se realiza la generación de subconjuntos de datos para el entrenamiento de árboles.
- Random Subspace: elección del sub-espacio de características de manera aleatoria dentro del nodo para desglosar las ramas del árbol de decisión.

3.3.1 Algoritmo de generación de bosque de árboles aleatorios

El algoritmo se basará en la generación de B árboles, a partir de B muestras extraídas de los datos de entrenamiento proporcionados, para de esta forma contar con un conjunto de árboles (Bosque de árboles).

Generación del Bosque de Árboles:

Desde $b=1$ hasta $b=B$

Extraer una muestra Bootstrap de los datos de entrenamiento, de tamaño P

Generar un árbol de decisión T_b de la muestra Bootstrap, repitiendo los siguientes pasos hasta alcanzar el nodo terminal con base al criterio de paro requerido:

Se selecciona m variables de la p variables

Elegir cuál de las m variables nos da una mejor separación de las clases, dando el mejor punto de corte

Cortar el nodo con base a la mejor variable para obtener dos hijos

Se tiene un conjunto de B árboles de decisión, que son utilizados para realizar la clasificación.

El algoritmo primero pide se saque una muestra bootstrapping de los datos de entrenamiento. Es decir se obtendrá una muestra usando remplazo de los datos originales de entrenamiento, con este subconjunto se determinará un árbol de decisión siguiendo un mecanismo similar al mostrado en la subsección anterior. En realidad se harán B árboles de decisión, todos con muestras diferentes. Cada uno de estos árboles representa un clasificador débil. La decisión final sobre una nueva muestra será aquella con mayor número de votos por parte de estos clasificadores débiles. El acumulado de estos clasificadores débiles representa un clasificador más robusto.

3.4 Validación cruzada

Este método es utilizado cuando la información con la que se cuenta no es la suficiente como para tener datos de entrenamiento y de prueba. Estos datos de entrenamiento son la entrada que se le da a la máquina de aprendizaje ya sea Random Forest, Árboles de decisión, Support Vector Machine, para obtener un modelo y evaluarlo con datos de entrenamiento de los cuales sabemos cuál es la respuesta.

La validación cruzada es la estratificación de todo el corpus que se tiene, y este se divide en partes iguales, el caso más común es la división de los datos en 10 partes iguales donde de manera secuencial utilizaremos 9 de las partes como datos de entrenamiento y 1 parte de prueba. Con lo anterior tendremos 10 secciones de datos de entrenamiento diferentes y sus 10 partes de datos de entrenamiento, donde por turnos se hará la prueba sobre todos los datos que poseemos.

Una vez finalizado el proceso de validación cruzada se habrá evaluado todo el corpus con el que se contaba, a continuación se mostrará una imagen donde se representará el proceso de la validación cruzada:

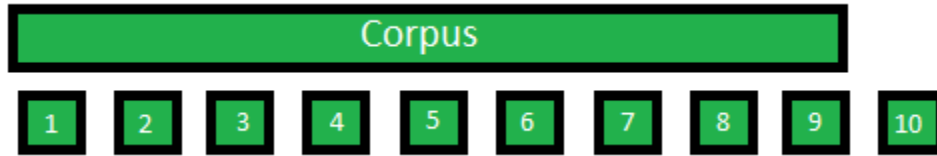


Figura 17 Particionamiento del corpus en diez secciones.

El primer paso es para realizar la validación cruzada es una vez que se tiene el corpus realiza la estratificación de éste, para la imagen anterior se observa que se estratifica en 10 partes.

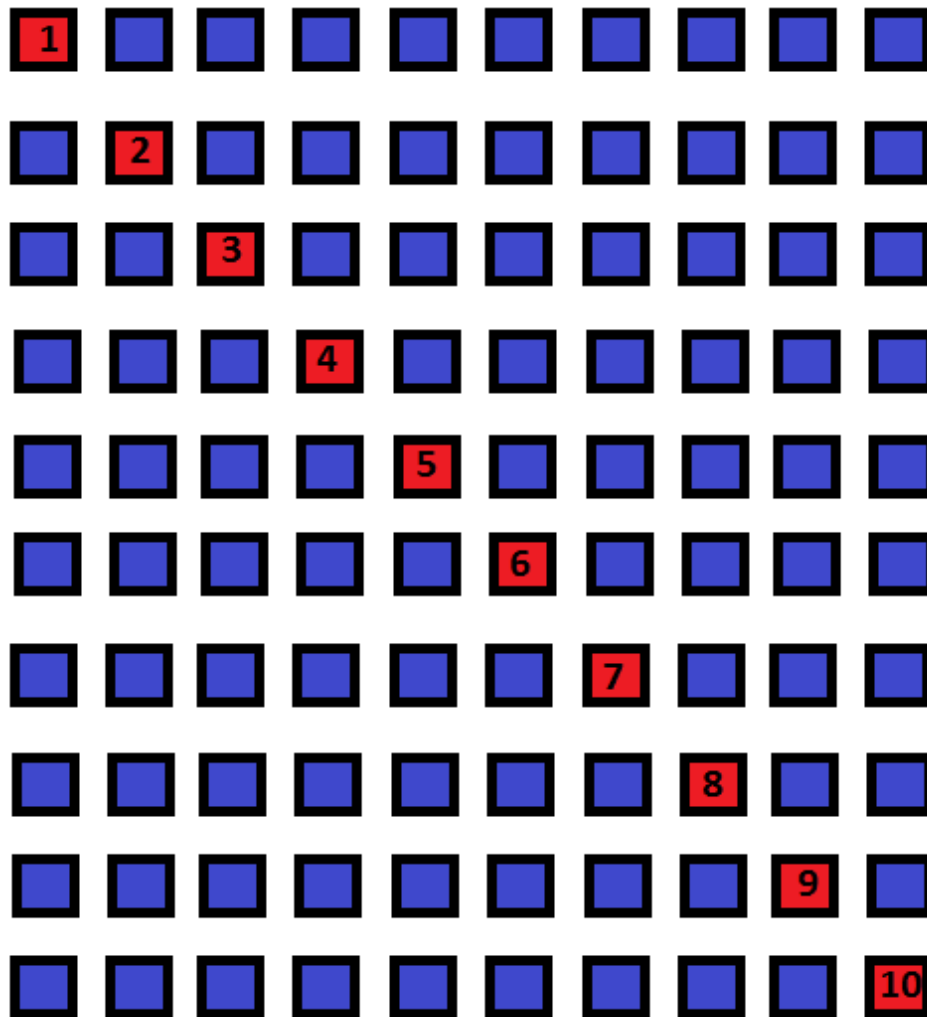


Figura 18 Proceso de iteración de las estratificaciones del corpus.

El proceso de iteración es el más importante, ya que ya que por cada iteración una estratificación del corpus es utilizado como datos de prueba (Color rojo) mientras que los 9 restantes son utilizados como datos de entrenamiento (Color azul), como se observa en la figura anterior se observa que el paso anterior se repite 10 veces hasta que todo el corpus es utilizado como datos de prueba logrando así evaluar los 10 modelos generados durante las 10 iteraciones.

3.5 Máquinas de soporte vectorial

Son un conjunto de algoritmos de aprendizaje. Dado un conjunto de ejemplos de entrenamiento (muestras) podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Intuitivamente, una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases a 2 espacios lo más amplios posibles mediante un hiperplano de separación definido como el vector entre los 2 puntos, de las 2 clases, más cercanos al que se llama vector soporte. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de los espacios a los que pertenezcan, pueden ser clasificadas a una o la otra clase.

Una máquina de aprendizaje es un algoritmo de aprendizaje automático supervisado, la cual comprende de la máquina de soporte vectorial para la clasificación (SVC) y la máquina de soporte vectorial para la regresión (SVR). (numerentur)

Capítulo IV

Metodología

Conocer la arquitectura del sistema, los métodos utilizados, el número de géneros que fueron utilizados. En este capítulo se presentará de manera detallada como es que se implementaron las herramientas mencionadas en el capítulo anterior, a su vez como la estructura de los elementos con que se desarrolló este trabajo.

4.1 Base de datos de género.

La base de datos de género es la recopilación que se obtuvo de los diferentes géneros a trabajar este proyecto, el cual se podría definir como un conjunto de entrenamiento. Con el cual se ira trabajando.

Debido a que hay muchos géneros musicales en este trabajo el campo de estudio se redujo a 4 géneros los cuales son:

- Balada (pop)
- Cumbia
- Rock
- Salsa

Para identificar los géneros musicales se utilizará una máquina de aprendizaje automático, en particular Random Forest. Se hará la recolección y etiquetado de las muestras de cada género. En este caso el etiquetado consistirá en separación de los archivos de audio con base al género musical que contenga.

El corpus se encuentra compuesto de la siguiente manera:

Género musical	Total de canciones
cumbia	221
pop	466
rock	401
ruido	166
salsa	186

Tabla 3 Contenido del corpus de datos con la distribución de los géneros musicales.

4.2 Arquitectura del sistema.

Mediante una máquina de aprendizaje automático se hará un modelo de aprendizaje, basado en la información obtenida al aplicar los MFCC's, en donde previamente se clasificarán cada uno de los archivos que formarán parte de los ejemplos que se le darán.

Una vez obtenido el modelo se podrá desarrollar un sistema el cual hará uso de este modelo para poder analizar archivos. Para posteriormente poder identificar elementos en vivo para determinar a cuál de los géneros musicales pertenece a partir de lo que escuchará.

A continuación, se mostrará la arquitectura del sistema de entrenamiento:

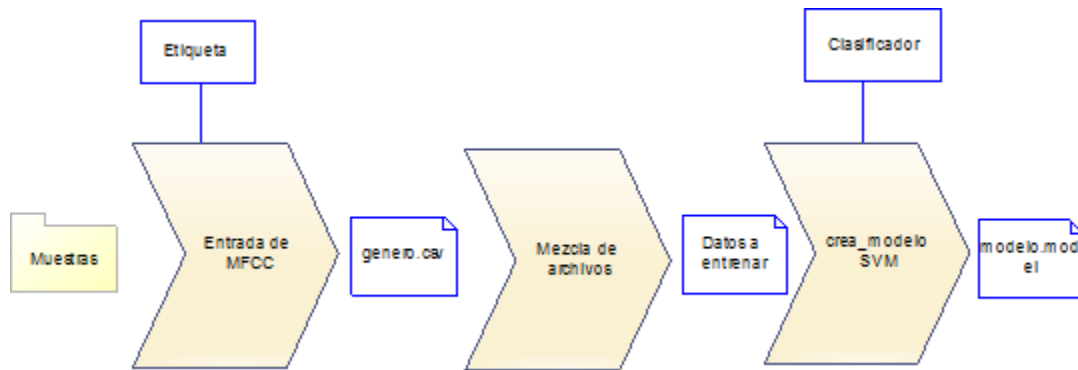


Figura 19 Sistema de entrenamiento

Primero se requiere tener las muestras de audio en formato .wav de un género, para que se genere un archivo de CSV (Coma Separated Values), el cual almacenará los valores de las estadísticas de los MFCC's de cada una de las ventanas de 10 segundos asociándolas a una etiqueta que indica el género de la canción y un identificador de la canción a la que pertenece.

Una vez que se genere el archivo con las estadísticas de los MFCC's de todos los géneros a analizar, procederemos a utilizar el Modulo "Mezcla de archivos", para generar los datos que se utilizaran para entrenar, "Datos a entrenar". Finalmente utilizaremos "Datos a entrenar" en el módulo "crea_modelo SVM", para poder generar un modelo, "modelo.model", que se ocupará posteriormente.

Con el modelo "modelo.model", generado en el sistema de entrenamiento determinaremos el género de un archivo de audio, donde nuestro módulo "Sistema identificador en archivo", requerirá del archivo a analizar y del modelo, para poder determinar el género de dicho archivo.



Figura 20 Sistema identificador de archivos

Ya que fue probado el modelo, “modelo.model”, mediante un archivo de audio previamente seleccionado, lo podremos utilizar en un sistema de identificación en vivo, para así poder evaluar cómo funciona de las dos maneras. La arquitectura de este sistema es la siguiente:



Figura 21 Sistema identificador en vivo

En la arquitectura podemos observar que el módulo “Sistema identificador en vivo”, requiere de dos elementos, la señal de audio la cual procesará se requiere de un aproximado de 10 segundos sin la parte introductoria de la canción, ya que en ocasiones es silencio y combinación de ritmos. Y el Modelo,” modelo.model” previamente generado; lo cual nos ayudara a determinar el “Género identificado”.

Capítulo V

Experimentos y resultados

En este capítulo se presentarán los escenarios en los que se realizaron los experimentos, en los que se extrajo el corpus de archivos de texto y se validó la clasificación de estos, la máquina de aprendizaje utilizada en todos los experimentos fue Random Forest.

5.1 Experimentos.

En este trabajo se realizaran tres experimentos los cuales serán uno con un corpus desbalanceado, otro con un corpus balanceado y uno extra el cual es realizado con un corpus externo al que se tendrá para los experimentos, este proporcionado por el trabajo realizado durante el verano científico en el IIMAS.

Durante el desarrollo de la tesis se realizó la obtención de archivos de audio de música, en 5 categorías distribuidas de la siguiente manera:

Género musical	Total de canciones
cumbia	221
pop	466
rock	401
ruido	166
salsa	186

Tabla 4 Base de datos con generos desbalanceados

Con este corpus se realizaron dos experimentos, uno en el que se tiene el corpus original y otro en el que se balancearon el número de canciones.

5.1.1 Experimento con corpus desbalanceado

El contenido del corpus de entrenamiento fue el corpus original, con las estadísticas de los MFCC's de ventanas de 10 segundos de las 1,440 canciones.

Este corpus se encuentra almacenado en un archivo de texto en el que se cuentan con las estadísticas de la ventana de 10 segundos de audio, el id de la canción a la que pertenece y el género al que pertenece. En el experimento se realizará utilizando la máquina de aprendizaje Random Forest, generando 5 árboles distintos con una profundidad máxima de 117 niveles, aplicando validación cruzada con 10 iteraciones.

Las métricas Precision, Recall y F-mesasures, son calculadas con base a la clasificación por canción, la clasificación de la canción se basa verificar a que genero pertenecen la mayoría de las ventanas de 10 segundos analizadas.

Matriz de confusión

Una matriz de confusión es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las

instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases.

Resultado del experimento con corpus desbalanceado

Tp	Tn	Fp	Fn		
210188	12456	92842	103465		
EVALUACIÓN FINAL POR CANCIÓN					
Tp	Tn	Fp	Fn		
1011	54	344	361		
MATRIZ DE CONFUSIÓN					
	CUMBIA	SALSA	ROCK	POP	RUIDO
CUMBIA	83	22	6	100	10
SALSA	8	93	4	74	5
ROCK	0	0	389	437	2
POP	11	15	0	81	54
RUIDO	83	22	6	100	10

Tabla 5 Resultados de la evaluación del experimento con el corpus desbalanceado

Precision	75.24%
Recall	74.19%
F-Measure	74.71%

Tabla 6 Métricas de la evaluación del experimento con el corpus desbalanceado

En esta clasificación podemos observar que el género mejor clasificado es Rock al clasificar todas las canciones de este género dentro del mismo, por otro se clasificaron como Pop 255 canciones que no eran, pero esto puede deberse a que se encuentra desbalanceado el corpus al contar con mayor número de canciones para el género Pop y Rock.

5.1.2 Experimento con corpus balanceado

El contenido del corpus de entrenamiento, la extracción de 160 canciones al azar por género, teniendo de esta manera balanceado el corpus, con las estadísticas de los MFCC's de ventanas de 10 segundos de las 800 canciones.

Este corpus se encuentra almacenado en un archivo de texto en el que se cuentan con las estadísticas de la ventana de 10 segundos de audio, el id de la canción a la que pertenece y el género al que pertenece. En el experimento se realizará utilizando la máquina de aprendizaje Random Forest, generando 5 árboles distintos con una profundidad máxima de 117 niveles, aplicando validación cruzada con 10 iteraciones.

Las métricas Precision, Recall y F-mesasures, son calculadas con base a la clasificación por canción, la clasificación de la canción se basa verificar a que genero pertenecen la mayoría de las ventanas de 10 segundos analizadas.

Distribución de los géneros dentro del corpus.

Género musical	Total de canciones
cumbia	160
pop	160
rock	160
ruido	160
salsa	160

Tabla 7 Contenido de canciones del experimento con el corpus balanceado

Resultado del experimento con corpus balanceado

EVALUACIÓN FINAL POR VENTANA					
Tp	Tn	Fp	Fn		
93613	19759	55611	70706		
EVALUACIÓN FINAL POR CANCIÓN					
Tp	Tn	Fp	Fn		
460	108	186	222		
MATRIZ DE CONFUSIÓN					
	CUMBIA	SALSA	ROCK	POP	RUIDO
CUMBIA	78	41	5	21	14
SALSA	13	121	2	12	9
ROCK	0	0	157	0	0
POP	11	29	0	104	13
RUIDO	11	30	0	11	108

Tabla 8 Resultados de la evaluación del experimento con el corpus balanceado.

Precision	71.20%
Recall	67.44%
F-Measure	69.27%

Tabla 9 Métricas de evaluación del experimento con corpus balanceado.

En el corpus balanceado podemos observar dentro de la clasificación presentada en la matriz de confusión, se ve claramente cómo es que el género Rock se encuentra identificado perfectamente al clasificar todas las canciones de este género dentro del mismo y son pocos de otros géneros los que son clasificados erróneamente en esta clase.

Análisis de métricas entre los dos corpus

	Precision	Recall	F-measure
Balanceado	71.20%	67.44%	69.27%
Desbalanceado	75.24%	74.19%	74.71%

Tabla 10 Comparativo de las métricas de evaluación de los experimentos con el corpus balanceado y desbalanceado.

Se observa que en el experimento donde está desbalanceado se obtienen mejores métricas de evaluación, aunque en el experimento balanceado se observa que hay una mejor identificación del ruido.

5.1.3 Experimento con corpus externo

En este experimento se utilizará el corpus recolectado durante el programa de “El verano de investigación científica 2015”, con sede en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, UNAM, a cargo del supervisor Iván Vladimir Meza Ruiz.

En este programa se dio un curso de aprendizaje automático, enfocado al audio en el que los estudiantes participantes recolectaron un corpus de audio, el cual es el que se utilizará en este experimento el cual cuenta con el siguiente contenido:

Género musical	Total de canciones
Bachata	235
Country	209
Jazz	223
Flamenco	93
Ska	117
Reggae	125

Tabla 11 Contenido de canciones dentro del corpus externo.

Este experimento se realizó igual que los anteriores, utilizando la máquina de aprendizaje automático Random Forest, utilizando validación cruzada con 10 iteraciones y realizando la evaluación con base a la clasificación de las canciones en el corpus.

Resultado del experimento con corpus externo

EVALUACIÓN FINAL POR VENTANA						
Tp	Tn	Fp	Fn			
143896	0	76567	76567			
EVALUACIÓN FINAL POR CANCIÓN						
Tp	Tn	Fp	Fn			
706	0	286	286			
MATRIZ DE CONFUSIÓN						
	BACHATA	COUNTRY	JAZZ	FLAMENCO	SKA	RAGGAE
BACHATA	214	9	9	0	0	0
COUNTRY	9	179	13	0	6	0
JAZZ	11	17	186	1	4	0
FLAMENCO	5	14	29	45	0	0
RAGGAE	58	23	36	0	8	0

Tabla 12 Matriz de confusión de la validación cruzada del corpus externo.

	Bachata	Country	Jazz	Flamenco	Ska	Reggae
<i>Bachata</i>	214	9	9	0	0	0
<i>Country</i>	9	179	13	0	6	0
<i>Jazz</i>	11	17	186	1	4	0
<i>Flamenco</i>	5	14	29	45	0	0
<i>Ska</i>	3	25	6	0	82	0
<i>Reggae</i>	58	23	36	0	8	0

Tabla 13 Matriz de confusión del experimento con el corpus externo.

Precision	0.711694
Recall	0.711694
F-Measure	0.711694

Tabla 14 Resultados de métricas del experimento con el corpus externo.

En la matriz de confusión podemos observar cómo la clasificación de Bachata, Country y Jazz es mejores que Flamenco, Ska y Reggae, Los primero géneros estos tienen mayor número de clasificaciones correctas en proporción al total de canciones de cada género, mientras que los segundos géneros tienen menor proporción de clasificaciones correctas. En el caso de Reggae se tiene que ninguna canción fue clasificada en dicho género, lo cual muestra que el corpus está desbalanceado en tal género, a pesar de que no sea el género con menor número de canciones.

Nieto menciona que los resultados de este experimento utilizando métricas son prometedores debido a que tiene una presión del 71%.

También menciona que fue de suma importancia el entrenamiento del modelo mediante “cross-validation”, ya que con este permite evadir el uso de fragmentos o ventanas tanto para el entrenamiento como para las pruebas que se realizaron. (Guillermo, 2015).

Capítulo VI

Conclusiones

Como se puede ver dentro de los resultados de precisión de los tres experimentos con validación cruzada, éstos se encuentran por encima del 70% de precisión dentro de lo cual cabe destacar que el experimento que presento mayor valor fue el experimento 1 con el corpus recolectado, donde se ve que los géneros con mayor número de canciones se identificaron mejor que los géneros con menor número de canciones. Los géneros mejor clasificados fueron pop y rock.

Esto también debido a que los géneros de salsa y cumbia al tener parte de sus orígenes en los mismos ritmos, tienen mucha similitud entre ellos lo que complica un poco poder obtener una diferenciación más concreta referente a los otros dos géneros.

Por otro lado, tenemos el corpus externo, el cual se recolectó en el verano científico que se llevó a cabo en el IIMAS en el 2015, donde Frine Solano y Adrian Espino estuvieron coordinando el trabajo realizado en este programa. Durante esta participación se pudo observar que hay más métodos y nuevas herramientas, para el procesamiento de audio, así como las diferentes formas de abordar el problema.

Siguiendo parte del proceso se pudo observar que de igual manera la precisión del sistema es mayor al 70%. Este realizado con su propios corpus lo que nos deja ver que el manejo de esta metodología tiene una eficiencia alta.

El menor desempeño entre experimento balanceado vs no balanceado se debe a dos factores.

* Primero, a que para balancearlo se disminuyó el número de datos de entrenamiento, esto va a tener un efecto y muy probablemente es que sea en un clasificador más pobre que uno con más datos

* Segundo, como se puede apreciar de ambos experimentos el rock es uno de los géneros "más fácil" de clasificar, aun cuando los datos son balanceados, rock tiene el mejor desempeño. Cuando los datos están desbalanceados, de lo que más tenemos son canciones de rock, por lo que en promedio el comportamiento del clasificador tiene más chances de atinarle, lo que se ve reflejado en el desempeño final.

Como trabajo a futuro, se tiene como idea que el corpus de entrenamiento con el que se entrena el sistema cuente con ruido, ya que esto afecta de manera drástica al sistema al momento de realizar el experimento en vivo, dado que el corpus de entrenamiento no tiene ruido, causando que el sistema presente dificultad de identificación de género del audio que escuche por el ruido presente en el ambiente.

Así como contar con un número equivalente de cada uno de los géneros, o en su caso utilizar a la par otra metodología para poder ver si el nivel de precisión podría aumentar y así poder realizar experimentos con un número mayor de géneros. Por ejemplo, metodologías de aprendizaje profundo que han mostrado gran avance en los últimos años.

Apéndice

Código utilizado

Crea_modelo.py

Este código permite crear el modelo para el entrenamiento del sistema.

```
import numpy as np
import scikits.audiolab as audiolab
import argparse
import pyaudio
import time
import numpy as np
import sys
import pickle
import os.path
import matplotlib.pyplot as plt
import matplotlib.image as mimage
import Image
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import NuSVC
parent=os.path.abspath('../..')
sys.path.append(parent)
from audiofun import mel
from scipy.fftpack import dct
from audiofun import mfc
from ef2csvlines import get_feats

def verbose(*args):
    print >> sys.stderr, "".join([str(x) for x in args])
parser = argparse.ArgumentParser()
parser.add_argument("entrada", type=str,
                    help="Archivo CSV")
parser.add_argument("salida", type=str,
                    help="archivo modelo")
opts = parser.parse_args()
archivo=open(opts.entrada,'r')
sal=open(opts.salida,'w')
valores=[]
categoria=[]
arraytotal=[]
```

```

renglones=0
##Convierte la cadena a diccionarios
for linea in archivo:
    if renglones !=0:
        linea=linea.replace(","," ")
        linea=linea.replace("cumbia","1")
        linea=linea.replace("pop","2")
        linea=linea.replace("salsa","3")
        linea=linea.replace("rock","4")
        linea=linea.replace("ruido","5")
        line=linea.strip()
        bits=line.split()
        elemento=[]
        for numero in bits:
            elemento.append(float(numero))
        arraytotal.append(elemento)
    renglones=renglones+1

for renglon in arraytotal:
    print renglon[117],renglon[116]
    valores.append(renglon[:117])
    categoria.append(renglon[117])
##Fin de la conversion

X = np.array(valores)
y = np.array(categoria)
print 'Empezando a entrenar'

#clf = NuSVC(nu=0.5, kernel='rbf', degree=3, gamma=0.0, coef0=0.0, shrinking=True, probability=False,
tol=0.001, cache_size=200, verbose=False, max_iter=-1, random_state=None)
#clf = NuSVC()
clf = RandomForestClassifier(n_estimators=10)
clf.fit(X, y)
print 'Termine de entrenar'
#Comprobar el entrenamiento
#for valor in valores:
#    print(clf.predict(valor))
gmms_=pickle.dumps(clf)
verbose('Saving model into ',opts.salida)
with open(opts.salida,"w") as model:
    model.write(gmms_)

```

1_sonidos_a_csv_ventana_id.py

Este código permite generar un archivo de csv para una carpeta de Canciones del mismo género.

```
#!/usr/bin/env python
#-*- coding: utf-8
import numpy as np
import scikits.audiolab as audiolab
import argparse
import pyaudio
import time
import numpy as np
import sys
import os.path
import matplotlib.pyplot as plt
import matplotlib.image as mimage
import Image
from time import time
parent=os.path.abspath('../..')
sys.path.append(parent)
from audiofun import mel
from scipy.fftpack import dct
from audiofun import mfc
from ef2csvlines import get_feats
parser = argparse.ArgumentParser()
parser.add_argument("samples", type=str,
                    help="Directorio a analizar")
parser.add_argument("categoria", type=str,
                    help="Categoria a analizar")
parser.add_argument("archivo", type=str,
                    help="Archivo a escribir")
parser.add_argument("num", type=int,
                    help="Valor inicial del id")
opts = parser.parse_args()
archiCSV=open(opts.archivo,'wb')
listing=os.listdir(opts.samples)
listing=["{0}/{1}".format(opts.samples,namefile)
        for namefile in listing
        if namefile.endswith('wav')
        or namefile.endswith('WAV')]
identificador=opts.num
vector_mffc=[]
for filename in listing:
    identificador=identificador+1
    start_time = time()
    sndf=audiolab.Sndfile(filename,'r')
    n=sndf.nframes
    data=sndf.read_frames(n)
    seg_totales=int(n/44100)
```

```

seg_validos=seg_totales-9
segundos=range(seg_validos)
factor=[]
for seg in segundos:
    ini=(seg)*44100
    fin=(seg+10)*44100
    factor.append([ini,fin])

#Rango de los datos que se analizaran
#print factor

if len(data.shape)>1:
    data=data[:,0]
numero=0
contador=0
for rango in factor:
    numero +=1
    inicio = rango[0]
    final = rango[1]
    datos=data[inicio:final]
    feats=get_feats(datos,mel.MELfilterbank_speech,mode="turn",feats=["mfcc"],
stats=["mean","std","q1","q2","q3",\
"min","max","skew","kurtosis"])
    feats.append(identificador)
    vector_mfcc.append(feats)
    sndf.close()
    elapsed_time = time() - start_time
    print filename,seg_totales, ' Segundos'
print("Tiempo de revision: %0.10f seconds." % elapsed_time)
column=""
for i in range(117):
    col='I'+str(i)+''
    column=column+col
column=column+'id,cat\n'
cadena=str(vector_mfcc)
cadena=cadena.replace(' ', [';','+opts.categoria+"\n')
cadena=cadena.replace(';', [';','+opts.categoria+"\n')
cadena=cadena.replace('I','')
cadena=cadena.replace('I','')
column=column+cadena+' '+opts.categoria
archiCSV.write(column)

```

2_archivo_csv_final.py

Este código permite generar un archivo de csv a partir de varios archivos csv

```
#!/usr/bin/env python
#-*- coding: utf-8
import numpy as np
import argparse
import pyaudio
import time
import numpy as np
import sys
import os.path
parent=os.path.abspath('../..')
sys.path.append(parent)
from audiofun import mel
from scipy.fftpack import dct
from audiofun import mfc
from ef2csvlines import get_feats
parser = argparse.ArgumentParser()
parser.add_argument("samples", type=str,
                    help="Directorio a analizar de csv")
parser.add_argument("archivo", type=str,
                    help="Archivo a escribir con los csv finales")
opts = parser.parse_args()
archiCSV=open(opts.archivo,'wb')
listing=os.listdir(opts.samples)
listing=["{0}/{1}".format(opts.samples,namefile)
         for namefile in listing
         if namefile.endswith('csv')
         or namefile.endswith('CSV')]
archivo=""
cabeza=""
for filename in listing:
    archi=open(filename,'r')
    contado=0
    for linea in archi:
        if contado ==0:
            cabeza=linea
        else:
            archivo=archivo+linea
        contado+=1
    archi.close()
archivo=cabeza+archivo
archiCSV.write(archivo)
```


3_evaluacion_v8_id.py

Este código permite tener la evacuación del sistema.

```
#!/usr/bin/env python
#-*- coding: utf-8
#Este codigo lee un archivo de csv, y realiza la evaluacion
#realizando validacion cruzada.
import csv
import random
import argparse
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import NuSVC
import numpy as np
from collections import OrderedDict
##Guardara los valores del porcentaje de acierto de cada cancion en la evaluacion
counters=[]
##validacion
tp=0
fp=0
tn=0
fn=0
##
tp_cancion=0
fp_cancion=0
tn_cancion=0
fn_cancion=0
##
cumbia_verdaderos=0
salsa_verdaderos=0
pop_verdaderos=0
rock_verdaderos=0
ruido_verdaderos=0
buenos=0
####
cumbia_falsos=0
salsa_falsos=0
pop_falsos=0
rock_falsos=0
ruido_falsos=0
##
iii=0
##Funcion que separa los features de la etiqueta
def separa_etiqueta(lista):
    lista_sola=[]
    etiquetas=[]
    for registro in lista:
        elemento=[]
        if len(registro)==119:
```

```

        for row in registro:
            if row != registro[-1]:
                elemento.append(row)
            else:
                etiquetas.append(row)
        lista_sola.append(elemento)
    return lista_sola,etiquetas

def evalua(vec_dtest,vec_etest,vec_dtrain,vec_etrain):
    global tp,tn,fp,fn,iii,tp_cancion,tn_cancion,fp_cancion,fn_cancion
    clf = RandomForestClassifier(n_estimators=10)
    vec_dtrain = np.array(vec_dtrain)
    vec_etrain = np.array(vec_etrain)
    clf.fit(vec_dtrain[:,range(0,117)], vec_etrain)
    pos=0
    #Inicializamos antes de evaluar a cada caso
    tp_id=0
    tn_id=0
    fp_id=0
    fn_id=0

    cumbia_id=0
    salsa_id=0
    pop_id=0
    rock_id=0
    ruido_id=0
    #Contador de registros con un mismo identificador
    reg_id=0
    ##Primer intento
    primeravez=True
    ##MATriz datos por registro
    metricas_registro=[]
    cadena='id,total,tp,tn,fp,fn\r\n'
    ##Inicializamos el valor anterior a un valor que no existe dentro de los ids
    anterior=0
    pos_id=0
    for datos in vec_dtest:
        ##Recolectamos el id del registro
        actual=datos[117]

        if primeravez:
            anterior=actual
            primeravez=False
        if anterior!=actual:
            ##(id_cancion,)
            metricas_registro.append((anterior,pos_id,tp_id,tn_id,fp_id,fn_id))
            genero=vec_etest[pos-1]
            maximo=max(cumbia_id,salsa_id,pop_id,rock_id,ruido_id)
            genero_actual=""
            if cumbia_id ==maximo:
                genero_actual="cumbia"
            elif salsa_id ==maximo:

```

```

        genero_actual="salsa"
    elif rock_id ==maximo:
        genero_actual="rock"
    elif pop_id ==maximo:
        genero_actual="pop"
    elif ruido_id ==maximo:
        genero_actual="ruido"
    else:
        print 'Genero no identificado'
    if genero_actual== genero:
        if genero_actual == "ruido":
            tn_cancion=tn_cancion+1
        else:
            tp_cancion=tp_cancion+1
    else:
        if genero_actual == "ruido":
            fn_cancion=fn_cancion+1
        else:
            fp_cancion=fp_cancion+1
            fn_cancion=fn_cancion+1

```

```

cadena=cadena+str(anterior)+','+str(pos_id)+','+str(tp_id)+','+str(tn_id)+','+str(fp_id)+','+str(fn_id)+','+str(genero_actual)+"\r\n"

```

```

    tp_id=0
    tn_id=0
    fp_id=0
    fn_id=0

    cumbia_id=0
    salsa_id=0
    pop_id=0
    rock_id=0
    ruido_id=0

    pos_id=0
    anterior=actual

```

```

respuesta = clf.predict(np.array(datos)[0:117])
#cadena=cadena+"\n"+str(vec_etest[pos])+','+str(respuesta)
    if respuesta == "cumbia":
        cumbia_id=cumbia_id+1
    elif respuesta == "salsa":
        salsa_id=salsa_id+1
    elif respuesta == "rock":
        rock_id=rock_id+1
    elif respuesta == "pop":
        pop_id=pop_id+1
    elif respuesta == "ruido":
        ruido_id=ruido_id+1
    else:
        print 'Error de clasificacion.'
if vec_etest[pos]== respuesta:
    if respuesta == "ruido":

```

```

        tn=tn+1
        tn_id=tn_id+1
    else:
        tp+=1
        tp_id+=1
else:
    if respuesta == "ruido":
        fn=fn+1
        fn_id=fn_id+1
    else:
        fp=fp+1
        fn=fn+1
        fp_id=fp_id+1
        fn_id=fn_id+1
    anterior=actual
    pos+=1
    pos_id+=1
archivo_metricas=open('archivo_metricas_'+str(iii)+'.txt','w')
archivo_metricas.write(cadena)
archivo_metricas.close()
iii=iii+1

```

```

def crea_idprueba(listaid, cuerpo):
    resultado=[]
    for item in listaid:
        res= [X for X in list(cuerpo) if (X[-2] ==item)]
        resultado=resultado+res
    return resultado
parser = argparse.ArgumentParser()
parser.add_argument("entrada", type=str,
                    help="Archivo CSV")
opts = parser.parse_args()

```

```

corpus = []
parte1 = []
parte2 = []
parte3 = []
parte4 = []
parte5 = []
parte6 = []
parte7 = []
parte8 = []
parte9 = []
parte10 = []
renglon=0
with open(opts.entrada, 'rb') as csvfile:
    spamreader = csv.reader(csvfile, delimiter=',', quotechar='\n')
    for row in spamreader:
        corpus.append(row)

```

##Eliminamos las etiquetas del archivo

```

corpus = corpus[1:]
tam=int(len(corpus) /10)
#Revolvemos la lista
random.shuffle(corpus)
print "Corpus mezclado"
##Pasar a flotante los numeros

datos=[]
ids=[]
for registro in corpus:
    copia=[]
    if len(registro)==119:
        for columna in registro:
            #Se convierte a flotante
            if columna != registro[-1]:#Descartar el genero de la cancion
                if columna != registro[-2]:#Descartar el id de la cancion
                    copia.append(float(columna))
                else:
                    copia.append(int(columna))
                    ids.append(int(columna))
            else:
                copia.append(columna)
        datos.append(copia)
##Eliminamos los ids repetidos dejando solo uno de ellos
ids=list(OrderedDict.fromkeys(ids))
tam_ids=tam=int(len(ids) /10)
#mezclamos los ids
random.shuffle(ids)
#REvolvemos los identificadores
print"Revolvi los ids"
corpus=datos
print "Corpus normalizado"

###Particion de los Ids
##Particion de los Ids
id1=ids[0:tam_ids]
ids=ids[tam_ids:]
id2=ids[0:tam_ids]
ids=ids[tam_ids:]
id3=ids[0:tam_ids]
ids=ids[tam_ids:]
id4=ids[0:tam_ids]
ids=ids[tam_ids:]
id5=ids[0:tam_ids]
ids=ids[tam_ids:]
id6=ids[0:tam_ids]
ids=ids[tam_ids:]
id7=ids[0:tam_ids]
ids=ids[tam_ids:]
id8=ids[0:tam_ids]
ids=ids[tam_ids:]
id9=ids[0:tam_ids]

```

```

ids=ids[tam_ids:]
id10=ids[0:tam_ids]
ids=ids[tam_ids:]
if len(ids)>0:
    id1.append(ids.pop())
if len(ids)>0:
    id2.append(ids.pop())
if len(ids)>0:
    id3.append(ids.pop())
if len(ids)>0:
    id4.append(ids.pop())
if len(ids)>0:
    id5.append(ids.pop())
if len(ids)>0:
    id6.append(ids.pop())
if len(ids)>0:
    id7.append(ids.pop())
if len(ids)>0:
    id8.append(ids.pop())
if len(ids)>0:
    id9.append(ids.pop())
##

#1
i=0
parte1=crea_idprueba(id1, corpus)
print "arreglo terminado No.1"
#2
i=1
parte2=crea_idprueba(id2, corpus)
print "arreglo terminado No.2"
#3
i=2
parte3=crea_idprueba(id3, corpus)
print "arreglo terminado No.3"
#4
i=3
parte4=crea_idprueba(id4, corpus)
print "arreglo terminado No.4"
#5
i=4
parte5=crea_idprueba(id5, corpus)
print "arreglo terminado No.5"
#6
i=5
parte6=crea_idprueba(id6, corpus)
print "arreglo terminado No.6"
#7
i=6
parte7=crea_idprueba(id7, corpus)
print "arreglo terminado No.7"
#8

```

```

i=7
parte8=crea_idprueba(id8, corpus)
print "arreglo terminado No.8"
#9
i=8
parte9=crea_idprueba(id9, corpus)
print "arreglo terminado No.9"
#10
i=9
parte10=crea_idprueba(id10, corpus)
print "arreglo terminado No.10"
##Caso 1
corpus_test=parte1
corpus_train=parte2+parte3+parte4+parte5+parte6+parte7+parte8+parte9+parte10
valores_test,etiquetas_test=separa_etiqueta(corpus_test)
valores_train,etiquetas_train=separa_etiqueta(corpus_train)
evalua(valores_test,etiquetas_test,valores_train,etiquetas_train)
print '10%'

##Caso 2
corpus_test=parte2
corpus_train=parte1+parte3+parte4+parte5+parte6+parte7+parte8+parte9+parte10
valores_test,etiquetas_test=separa_etiqueta(corpus_test)
valores_train,etiquetas_train=separa_etiqueta(corpus_train)
evalua(valores_test,etiquetas_test,valores_train,etiquetas_train)
print '20%'

##Caso 3
corpus_test=parte3
corpus_train=parte2+parte1+parte4+parte5+parte6+parte7+parte8+parte9+parte10
valores_test,etiquetas_test=separa_etiqueta(corpus_test)
valores_train,etiquetas_train=separa_etiqueta(corpus_train)
evalua(valores_test,etiquetas_test,valores_train,etiquetas_train)
print '30%'

##Caso 4
corpus_test=parte4
corpus_train=parte2+parte3+parte1+parte5+parte6+parte7+parte8+parte9+parte10
valores_test,etiquetas_test=separa_etiqueta(corpus_test)
valores_train,etiquetas_train=separa_etiqueta(corpus_train)
evalua(valores_test,etiquetas_test,valores_train,etiquetas_train)
print '40%'

##Caso 5
corpus_test=parte5
corpus_train=parte2+parte3+parte4+parte1+parte6+parte7+parte8+parte9+parte10
valores_test,etiquetas_test=separa_etiqueta(corpus_test)
valores_train,etiquetas_train=separa_etiqueta(corpus_train)
evalua(valores_test,etiquetas_test,valores_train,etiquetas_train)
print '50%'
##Caso 6
corpus_test=parte6

```

```
corpus_train=parte2+parte3+parte4+parte5+parte1+parte7+parte8+parte9+parte10
valores_test,etiquetas_test=separa_etiqueta(corpus_test)
valores_train,etiquetas_train=separa_etiqueta(corpus_train)
evalua(valores_test,etiquetas_test,valores_train,etiquetas_train)
print '60%'
```

```
##Caso 7
corpus_test=parte7
corpus_train=parte2+parte3+parte4+parte5+parte6+parte1+parte8+parte9+parte10
valores_test,etiquetas_test=separa_etiqueta(corpus_test)
valores_train,etiquetas_train=separa_etiqueta(corpus_train)
evalua(valores_test,etiquetas_test,valores_train,etiquetas_train)
print '70%'
```

```
##Caso 8
corpus_test=parte8
corpus_train=parte2+parte3+parte4+parte5+parte6+parte7+parte1+parte9+parte10
valores_test,etiquetas_test=separa_etiqueta(corpus_test)
valores_train,etiquetas_train=separa_etiqueta(corpus_train)
evalua(valores_test,etiquetas_test,valores_train,etiquetas_train)
print '80%'
```

```
##Caso 9
corpus_test=parte9
corpus_train=parte2+parte3+parte4+parte5+parte6+parte7+parte8+parte1+parte10
valores_test,etiquetas_test=separa_etiqueta(corpus_test)
valores_train,etiquetas_train=separa_etiqueta(corpus_train)
evalua(valores_test,etiquetas_test,valores_train,etiquetas_train)
print '90%'
```

```
##Caso 10
corpus_test=parte10
corpus_train=parte2+parte3+parte4+parte5+parte6+parte7+parte8+parte9+parte1
valores_test,etiquetas_test=separa_etiqueta(corpus_test)
valores_train,etiquetas_train=separa_etiqueta(corpus_train)
evalua(valores_test,etiquetas_test,valores_train,etiquetas_train)
print '100%'
```

```
##Lineas de validacion
print '\nEvaluacion Final'
print 'Tp = ',tp,' Tn = ',tn,' Fp = ',fp,' Fn = ',fn
print '\nEvaluacion Final'
print 'Tp = ',tp_cancion,' Tn = ',tn_cancion,' Fp = ',fp_cancion,' Fn = ',fn_cancion
```


Resultados

Resultado del experimento con corpus desbalanceado

```
Evaluacion Final por ventana
Tp = 210188    Tn = 12456    Fp = 92842    Fn = 103465

Evaluacion Final por cancion
Tp = 1011     Tn = 54     Fp = 344     Fn = 361

Matriz de confusion
      cumbia  salsa  rock  pop  ruido
"cumbia"  83    22    6    100  10
"salsa"   8    93    4    74   5
"rock"    0     0   398    0   0
"pop"     6    11    6   437   2
"ruido"   11   15    0    81  54
```

Figura 22 Resultados de la evaluación del experimento con el corpus desbalanceado

Resultado del experimento con corpus balanceado

```
Evaluacion Final por ventana
Tp = 93613    Tn = 19759    Fp = 55611    Fn = 70706

Evaluacion Final por cancion
Tp = 460     Tn = 108     Fp = 186     Fn = 222

Matriz de confusion
      cumbia  salsa  rock  pop  ruido
"cumbia"  78    41    5    21  14
"salsa"   13   121    2    12   9
"rock"    0     0   157    0   0
"pop"     11   29    0   104  13
"ruido"   11   30    0    11  108
```

Figura 23 Resultados de la evaluación del experimento con el corpus balanceado.

Resultado del experimento con corpus externo

```
Evaluacion Final por ventana
Tp = 143869   Tn = 0   Fp = 76567   Fn = 76567

Evaluacion Final por cancion
Tp = 706     Tn = 0     Fp = 286     Fn = 286

Matriz de confusion
      bachata country jazz   flamenco   ska   reggae
"bachata"  214    9    9    0    0    0    0
"country"   9    179  13    0    6    0
"jazz"     11    17  186    1    4    0
"flamenco"  5    14  29   45    0    0
"ska"       3    25    6    0   82    0
"reggae"    58   23   36    0    8    0
```

Figura 24 Matriz de confusión de la validación cruzada del corpus externo

Bibliografía

Trabajos citados

- ALICIA, R. B. (2007). *Música I ESO (LOMCE)*. ESPAÑA: EDITEX.
- Camacho González, A. (2015). *Caracterización de perfiles en Twitter con Random Forest. Licenciatura*. México : Universidad Nacional Autónoma de México, Facultad de Ciencias.
- Denizeau, G. (2005). *Los Géneros Musicales*. Barcelona España: RobinBook.
- Gerkey B, C. K. (2009). *ROS: An opensource Robot Operating System. In ICRA Workshop on Open Source Software*.
- Guervós, L. E. (2008). Nietzsche y la danza. I parte. *DANZA BALLETT LA DANZA Y EL BALLETT*.
- Guillermo, N. H. (2015). *Identificación de géneros musicales como habilidad para robot de servicio Golem III*. Estado de México, México.: Instituto Tecnológico de Toluca, Departamento de Ingeniería Eléctrica y Electrónica.
- Lahoz-Beltra, R. (2004). *Bioinformática: Simulación, vida artificial e inteligencia artificial*. Madrid: Diaz de Santos.
- Lim, S. C. (2012). Music-genre classification system based on spectro-temporal features and feature selection. *Consumer Electronics, IEEE Transactions on*, 58(4), 1262-1268.
- Lobo, M. R. (2003). *Música para todos. Una Introducción Al Estudio de la Música*. Costa Rica: Univerdidad de Costa Rica.
- Morales de Álava José María, U. J. (1999). *HOMO MUSICALIS*.
- Nicolas Scaringella, G. Z. (2006). Automatic Genre Classification of Music Content.
- numerentur. (s.f.). Maquinas de Soporte Vectorial (SVM). numerentur.org/svm/.
- Pineda, L. a. (2013). *The Golem Team, RoboCup@Home 2013. Proceedings of RoboCup*. México, D.F.
- Pineda, L. R. (s.f.). *Concept and Functional Structure of a Service Robot*. International Journal of Advanced Robotics Systems.
- Pineda, L. S. (s.f.). *SitLog: A Programming Language for Service Robots Tasks*. International Journal of Advanced Robotic Systems.

Rascón, C. M. (s.f.). *Integration of the Multi-DOA Estimation Functionality to Human-Robot Interaction*. International Journal of Advanced Robotic Systems.

Sánchez, J. A. (2002). *Avances en robótica y visión por computador*. España.

SuperRadio. (2013). *Historia y Origen de la Salsa*. SupermixRadio.net.

Tzanetakis, G. &. (2002). Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transaction on*, 293-302.