

**DISEÑO DE UN SOFTWARE DE AUDICIÓN COMPUTACIONAL
PARA LA CLASIFICACIÓN DE UN GÉNERO FOLCLÓRICO
COLOMBIANO DE LAS REGIONES ANDINA, ATLÁNTICA,
PACÍFICA Y ORINOQUÍA**

JOSÉ ESTEBAN ZAMORA ALBARRACÍN
Código: 162212147

UNIVERSIDAD DE CUNDINAMARCA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
FUSAGASUGÁ.

2019

**DISEÑO DE UN SOFTWARE DE AUDICIÓN COMPUTACIONAL
PARA LA CLASIFICACIÓN DE UN GÉNERO FOLCLÓRICO
COLOMBIANO DE LAS REGIONES ANDINA, ATLÁNTICA,
PACÍFICA Y ORINOQUÍA**

Trabajo de grado presentado como requisito para optar por el título de ingeniero
electrónico

JOSÉ ESTEBAN ZAMORA ALBARRACÍN
Código: 162212147

Director:
LEONARDO RODRÍGUEZ MÚJICA
Ingeniero electrónico

Codirector:
ARLEY FERNANDO FERNÁNDEZ
Ingeniero mecatrónico

Línea investigación:
**SOFTWARE, SISTEMAS EMERGENTES y NUEVAS
TECNOLOGÍAS**

**UNIVERSIDAD DE CUNDINAMARCA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
FUSAGASUGÁ.**

2019

DEDICATORIA

A mis padres, por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes he logrado llegar hasta aquí y convertirme en lo que soy. Ha sido el orgullo y el privilegio de ser su hijo, son los mejores padres.

AGRADECIMIENTOS

Quiero agradecer a la base de todo, a mi familia, en especial a mis padres y abuelos, que quienes con sus consejos fueron el motor de arranque y mi constante motivación, muchas gracias por su paciencia y comprensión, y sobre todo por su amor.

Así, quiero mostrar mi gratitud a todas aquellas personas que estuvieron presentes en la realización de esta meta, de este sueño que es tan importante para mí, agradecer todas sus ayudas, sus palabras motivadoras, sus conocimientos, sus consejos y su dedicación.

Muestro mis más sinceros agradecimientos a mi tutor de proyecto, quien con su conocimiento y su guía fue una pieza clave para que pudiera desarrollar una clave de hechos que fueron imprescindibles para cada etapa de desarrollo del trabajo.

RESUMEN

Este trabajo incorpora elementos a través de los cuales se pretende priorizar la observación, análisis e interpretación en la clasificación de la música folclórica colombiana, buscando normalizar las características relacionadas como la instrumentación o textura tímbrica; debido a ello, en este trabajo se concentra un alto componente cualitativo por parte de los métodos de aprendizaje. Este estudio iniciará con la indagación de dichas características de cuatro géneros folclóricos colombianos elegidos: Bambuco, Currulao, Joropo y Vallenato. Para estas muestras musicales se propone generar cuatro tópicos puntuales: primero, un vector de 9 características por canción el cual nos permite hacer una medición cuantitativa de las tipologías conocidas dentro de la textura tímbrica como: texturas brillantes, atenuación, cambio espectral de la señal, ruido de la señal y representación del habla; seguido de esto, se procederá a realizar la identificación de los diferentes algoritmos de aprendizaje automático con el análisis del contenido rítmico de la señal, donde se derivan buscando determinar el tipo de aprendizaje que se le debe aplicar al conjunto de datos, teniendo presente la cantidad de observaciones que ingresan al proceso de entrenamiento de la maquina y los parámetros que resaltan en cada método de aprendizaje; posteriormente, dentro del análisis de la tercer característica de los algoritmos de aprendizaje automático se busca establecer el método que presenta mejores resultados teniendo en cuenta características de optimización para trabajos futuros y Finalmente, generar una evaluación del método seleccionado donde estas mediciones con sus respectivas observaciones se hallan por supuesto teniendo en cuenta el estudio elaborado, sección bastante relevante en este trabajo, acerca de los inconvenientes que tiene y conlleva una clasificación y/o caracterización musical.

ABSTRACT

This work incorporates elements through which it is intended to prioritize the observation, analysis and interpretation in the classification of Colombian folk music, seeking to normalize related features such as instrumentation or timbral texture; due to this, in this work a high qualitative component is concentrated by the learning methods. This study will begin with the investigation of these characteristics of four Colombian folkloric genres chosen: Bambuco, Currulao, Joropo and Vallenato. For these musical samples we propose to generate four specific topics: first, a vector of 9 characteristics per song which allows us to make a quantitative measurement of the known typologies within the timbral texture such as: bright textures, attenuation, spectral change of the signal, signal noise and speech representation; followed by this, we will proceed to perform the identification of the different algorithms of machine learning with the analysis of the rhythmic content of the signal, where they are derived seeking to determine the type of learning that should be applied to the data set, keeping in mind the amount of observations that enter the training process of the machine and the parameters that stand out in each learning method; Later, in the analysis of the third characteristic of the automatic learning algorithms, we seek to establish the method that presents the best results, taking into account optimization characteristics for future work and, finally, generate an evaluation of the selected method where these measurements with their respective observations are they find, of course, taking into account the elaborated study, a section that is quite relevant in this work, about the inconveniences that it has and entails a classification and / or musical characterization.

Tabla de Contenidos

Índice de figuras	8
Índice de tablas	10
1. INTRODUCCIÓN	12
2. PLANTEAMIENTO DEL PROBLEMA	14
3. OBJETIVOS	16
3.1. Objetivo General	16
3.2. Objetivos Específicos	16
4. JUSTIFICACIÓN	17
5. ALCANCES Y LIMITACIONES	18
5.1. Alcances	18
5.2. Limitaciones	18
6. MARCO TEÓRICO	19
6.1. Fundamentos teóricos	19
6.1.1. Muestreo de las señales de audio	19
6.1.2. Transformada de la señal	21
6.1.3. Extracción de características	24
6.1.4. lenguaje R	29
6.1.5. Tipos de algoritmos de aprendizaje automático.	33
6.1.6. k Vecinos más cercanos	37
6.1.7. Clasificador Bayesiano ingenuo	40

6.1.8. SVM	42
6.1.9. Folclor Colombiano	48
6.2. ESTADO DEL ARTE	55
6.2.1. Clasificación a nivel mundial	55
6.2.2. Clasificación en Colombia	59
6.2.3. Cultura digital	61
7. DISEÑO METODOLÓGICO	64
7.1. Estandarización de los cuatro géneros folclóricos colombianos	64
7.1.1. Aproximación inicial	64
7.1.2. Audioteca	66
7.1.3. Banco de datos	66
7.2. Identificación los métodos de aprendizaje	76
7.2.1. Tabla comparativa métodos de aprendizaje	76
7.2.2. Comparación métodos de aprendizaje	78
7.3. Técnica para la clasificación	80
7.3.1. Inspección en la clasificación para la definición de la técnica	80
7.3.2. Velocidad de respuesta en las técnicas de aprendizaje	81
7.3.3. Verificación del proceso de clasificación	88
7.4. Evaluar porcentaje precisión del algoritmo	89
7.4.1. Matriz de confusión	89
8. RESULTADOS	92
8.1. Estandarización de los cuatro géneros folclóricos colombianos	92
8.1.1. Audioteca	92

8.1.2. Banco de datos	93
8.2. Identificación de los métodos de aprendizaje	97
8.2.1. Esquema comparativo métodos de aprendizaje	97
8.2.2. Comparación métodos de aprendizaje	101
8.3. Técnica para la clasificación	104
8.3.1. Inspección en la clasificación para la definición de la técnica	105
8.3.2. Velocidad de respuesta en las técnicas de aprendizaje	110
8.3.3. Verificación del proceso de clasificación	112
8.4. Evaluar porcentaje precisión del algoritmo	118
8.4.1. Clasificación géneros folclóricos	118
8.4.2. Clasificación géneros populares	121
9. CONCLUSIONES Y TRABAJOS FUTUROS	125
9.1. Conclusiones	125
9.2. Trabajos futuros	127
Referencias	128
ANEXO	131
A. Configuración youtube-dl	131
B. Algoritmo complemento extracción características	138
B.1. Octave	138
B.2. R	146
C. Tablas de resultado de clasificación	151

Índice de figuras

1.	Diagrama procesamiento de datos en R	31
2.	Hoja de trucos del algoritmo scikit-learn Python. tomado:	36
3.	Selección valor k . tomado:	39
4.	hiperplano SVM. tomado:	44
5.	Maximum Margin Hyperplane. tomado:	45
6.	Validación cruzada de K iteraciones. Fuente:	88
7.	Cabecera tabla muestras musicales	94
8.	Gráfico de dispersión características	98
9.	Características para clasificación en agrupación 3D	99
10.	Características para clasificación en agrupación 3D	99
11.	Gráfico de dispersión características MFCC	100
12.	Curva de aprendizaje modelo Knn.	101
13.	Curva de aprendizaje algoritmo clasificador bayesiano ingenuo.	102
14.	Curva de aprendizaje modelo SVM con kernel Radial.	103
15.	Modelo aprendizaje primer grupo.	106
16.	Modelo aprendizaje último grupo.	106
17.	Modelo aprendizaje primer grupo.	107
18.	Modelo aprendizaje último grupo.	107
19.	Modelo aprendizaje primer grupo.	108
20.	Modelo aprendizaje último grupo.	109
21.	Velocidad aprendizaje primer grupo.	110
22.	Velocidad aprendizaje último grupo.	111
23.	Velocidad aprendizaje primer grupo.	111

24.	Velocidad aprendizaje último grupo.	111
25.	Velocidad aprendizaje primer grupo.	112
26.	Velocidad aprendizaje último grupo.	112
27.	Validación cruzada 10 iteraciones grupo inicial	113
28.	Validación cruzada 10 iteraciones grupo final	113
29.	Validación cruzada 10 iteraciones grupo inicial	114
30.	Validación cruzada 10 iteraciones grupo final	114
31.	Validación cruzada 10 iteraciones grupo inicial	115
32.	Validación cruzada 10 iteraciones grupo final	115
33.	Densidad e identificación de características en el entrenamiento	116
34.	Densidad e identificación de características MFCC en el entrenamiento . . .	117
35.	Matriz de confusión algoritmo SVM.	119
36.	Matriz de confusión algoritmo SVM.	119
37.	Matriz de confusión algoritmo SVM.	120
38.	Matriz de confusión algoritmo SVM.	120
39.	Matriz de confusión algoritmo SVM.	121
40.	Matriz de confusión algoritmo SVM.	122
41.	Matriz de confusión algoritmo SVM.	122
42.	Matriz de confusión algoritmo SVM.	123

Índice de tablas

1.	Tabla lenguajes desarrollo aprendizaje de maquina	32
2.	Tabla cuantitativa lenguajes desarrollo aprendizaje de maquina	33
3.	Tabla Fortalezas y debilidades k-NN	38
4.	Tabla Fortalezas y debilidades Clasificador Bayesiano	42
5.	Tabla Fortalezas y debilidades SVM	47
6.	Tabla método de aprendizaje	77
7.	Matriz de confusión	90
8.	Nombres muestras musical Bambuco entrenamiento	94
9.	Nombres muestras musical Currulao entrenamiento	95
10.	Nombres muestras musical Joropo entrenamiento	95
11.	Nombres muestras musical Vallenato entrenamiento	96
12.	Porcentaje de acierto técnica manual	104
13.	Porcentaje de acierto métodos	109
14.	Nombres muestras musical clasificación géneros populares	124
15.	Nombres muestras musicales del Bambuco en la clasificación	152
16.	Nombres muestras musicales del Currulao en la clasificación	153
17.	Nombres muestras musicales del Joropo en la clasificación	154
18.	Nombres muestras musicales del Vallenato en la clasificación	155

Listings

1.	Centroide	67
2.	Atenuacion espectral	68
3.	Flujo espectral	69
4.	Cruce por cero	70
5.	Vector textura timbrica	71
6.	Textura timbrica	72
7.	Algoritmo de aprendizaje k-NN	81
8.	Algoritmo de aprendizaje Clasificador Bayesiano ingenuo	82
9.	Algoritmo de aprendizaje SVM	84
10.	Fragmento del algoritmo con el cual se grafican los diferentes modelos	85
11.	Comandos configuración youtube dl	92
12.	Ejemplos comandos youtube dl	131
13.	Comandos configuración youtube dl	131
14.	Comandos instalación curl youtube dl	132
15.	Comandos instalación wget youtube dl	132
16.	Ejemplos plantilla de salida	135
17.	Cell2csv	138
18.	cell2file	140
19.	Atenuacion espectral	140
20.	Centroide	141
21.	Cruce por cero	142
22.	Flujo espectral	143
23.	Textura timbrica	144
24.	Vector textura timbrica	146
25.	Algoritmo de aprendizaje k-NN	146
26.	Algoritmo de aprendizaje Clasificador Bayesiano ingenuo	148
27.	Algoritmo de aprendizaje SVM	149

1. INTRODUCCIÓN

El folclor colombiano tiene diversos géneros que identifican la cultura de cada región, de esta manera se generan múltiples estilos musicales dependientes de cada una, esta diversidad musical es originada por la mezcla de tres grandes comunidades como los indígenas americanos, los africanos y europeos, los cuales tienen una diversidad cultural y cada una de estas aportó la etnia para un verdadero mestizaje cultural, a todo esto se debe añadir el factor natural, donde el relieve colombiano agreste dominado por tres grandes cordilleras que han sido barreras físicas tan importantes impidiendo un intercambio cultural; permitiendo de esta manera mantener y desarrollar su propia cultura musical.

Estos diversos géneros, conforman y enriquecen la música folclórica colombiana, para diferenciar y caracterizar cada uno de ellos se tienen en cuenta la instrumentación o textura tímbrica, el contenido rítmico y el contenido armónico o tonal, a menudo esta caracterización musical se realiza manualmente. De manera que, el desarrollo de este proyecto es un trabajo investigativo y de aporte en la clasificación de los géneros musicales folclóricos colombianos, lo cual se considera un proceso subjetivo que se ve afectado por factores como la cultura, la localización geográfica, el autor. Por esto uno de los inconvenientes es no tener una forma incuestionable y única de clasificación. En ese orden de ideas inicialmente se realizará el estudio de los géneros Bambuco, Currulao, Joropo, Vallenato para determinar las características principales de cada uno de los géneros.

De manera que este trabajo está estructurado de la siguiente forma: la sección 1 es un marco referencial, donde se busca dejar claridad en cada uno de los términos utilizados, el lenguaje de programación usado en el transcurso de este proyecto y donde se le da la relevancia a los inconvenientes en la clasificación musical no solo nacional si no mundialmente hablando, teniendo presente el trabajo desarrollado por (Muñoz, 2016) donde realiza una aproximación musical folclórica colombiana mediante la extracción de las características. En la sección 2 se expone el diseño metodológico utilizado teniendo presente bajo qué criterios y parámetros se pretende realizar la caracterización de la música colombiana, la identificación de los métodos de aprendizaje, la selección del algoritmo mediante diferentes parámetros y por ultimo la evaluación del algoritmo seleccionado, esto ilustrado por cada uno de los algoritmos usados como herramientas para la realización de una clasificación como tal, así mismo mencionando su implementación a las canciones escogidas por cada uno de los cuatro géneros musicales colombianos, siendo un total de 64 canciones para el entrenamiento del algoritmo y 140 de clasificación y prueba del algoritmo para un total de 204 canciones clasificadas. Posteriormente, la sección 3 ilustra el análisis tanto cualitativo como cuantitativo del proceso realizado a nivel investigativo, haciendo por supuesto las

respectivas comparaciones de los modelos de aprendizaje donde se muestran los resultados conseguidos. Finalmente se logra culminar con unas conclusiones relevantes donde se obtiene que algoritmos de identificación grupal tienen mejor funcionamiento sea mediante métodos supervisados o no supervisados.

2. PLANTEAMIENTO DEL PROBLEMA

Los géneros musicales son categorías que han surgido ante la necesidad de organizar colecciones de música y caracterizar las similitudes entre músicos y composiciones. Pese a ello, los límites entre géneros siguen siendo difusos, así como su definición, haciendo que el problema de la clasificación no sea trivial y siendo aún más difícil la separación de subgéneros dentro de un mismo género. De la misma manera, los géneros tradicionales o folclóricos son transmitidos culturalmente sin intervención de la academia siendo parte de valor y cultura de un pueblo; en este contexto, mantener de alguna manera marcado el carácter étnico que normalmente dificulta comprender a escala internacional, teniendo algunas excepciones denominadas la música del mundo como el flamenco, el tango, la samba o el vallenato siendo este último, el género musical que la cultura colombiana aportó a la música del mundo según (Educational, 2015).

Por otro lado, es necesario categorizar los géneros musicales en comerciales e históricos, buscando conceptualizar, distinguir y definir de manera clara. Uno de los inconvenientes de clasificar los géneros por etiquetas es el proceso subjetivo que presenta la clasificación por los expertos, en lo cual el conocimiento personal y cultural puede influenciar en la forma en que da un veredicto, siendo una dificultad en la clasificación de grandes categorías de género culto o género popular, debido al alto grado de complejidad del conjunto, por lo cual es necesario buscar características comunes.

Siendo Colombia una de las culturas latinoamericanas más festivas, celebrando y disfrutando de las expresiones culturales y artísticas de sus pueblos, durante todo el año hay un variado calendario de ferias, festivales y carnavales que se realizan en todos los rincones del país, gran parte de los festivales que se realizan y organizan en Colombia están dedicados a la música folclórica, como una de las expresiones más importantes para los colombianos; cada año se celebran más de veinte festivales musicales en el país, en los cuales la diversidad étnica aporta ingredientes particulares a los centenares de viajeros nacionales e internacionales, generando la oportunidad de conocer de cerca la tradición y la multiculturalidad musical del país, la cual permite el desarrollo cultural y artístico de la población, donde festivales culturales de baja representación como el festival del arroz celebrado en el sur del Tolima, hasta grandes festividades nacionales como el festival de San pelayo, o el encuentro mundial del coleo, celebraciones donde la participación masiva de personas se expresa en el flujo de turismo cultural (*CO Colombia*, 2017) que permite mantener estas tradiciones y masificar las expresiones hasta tornar de estos en eventos internacionales.

Teniendo presente esto, la falta en el reconocimiento automático de los géneros folclóricos

donde la aproximación más cercana la plantea (García, Pacheco, y Mancera, 2005) en una clasificación de géneros populares y (Muñoz, 2016) en una aproximación en la caracterización de los géneros folclóricos colombianos, donde la existencia de las características que comparten entre sí, permite plantear el aprendizaje automático, el cual busca crear algoritmos de clasificación que le permitan a la maquina aprender y diferenciar patrones en los géneros a nivel nacional como bambuco, currulao, joropo y vallenato los cuales no han tenido una clasificación automática. Basado en estas declaraciones, y teniendo en cuenta que la clasificación de géneros musicales es subjetiva y depende de los aspectos culturales, temporales y personales surge la pregunta. ¿Es posible realizar la clasificación de los géneros folclóricos mediante herramientas como la audición computacional?

3. OBJETIVOS

3.1. Objetivo General

Diseñar un software de audición computacional que permita la clasificación de un género folclórico colombiano de las regiones Andina, Atlántica, Pacífica y Orinoquía

3.2. Objetivos Específicos

- Estandarizar los cuatro géneros folclóricos colombianos a nivel del vector de características.
- Identificar el tipo de aprendizaje adecuado para la clasificación de los cuatro géneros folclóricos colombianos a partir de sus características.
- Establecer una técnica para la clasificación de los cuatro géneros folclóricos colombianos a partir de sus características.
- Evaluar el porcentaje de precisión del algoritmo en la clasificación automática de los géneros folclóricos colombianos.

4. JUSTIFICACIÓN

En la actualidad la digitalización de piezas musicales mediante el reconocimiento de patrones y la audición por computadora ha permitido la organización de géneros populares en los cuales la cultura y la actividad comercial tiene un grado de participación alto, por otro lado, se ha generado una pérdida cultural que con el pasar de los años se ha tornado en grandes rasgos a la regeneración de los ritmos folclóricos, causando una clasificación digital nula de estos medios de expresión que no se han tenido presentes, para obtener una biblioteca digital en la cual se cataloguen los ritmos típicos que cada autor representa. Por esta razón el surgimiento de nuevos autores se ve seccionado en el reflejado de una cultura que cada vez se vuelve más popular que folclórica, impidiendo renovar o innovar en los diferentes géneros, además cerrando la participación en plataformas digitales por parte del folclor colombiano por la baja participación de autores en los diferentes ritmos.

Conforme a lo anterior, el presente estudio, tiene como finalidad contribuir con una herramienta de audición computacional, como soporte en la clasificación de géneros folclóricos colombianos, identificándolos de manera supervisada mediante el aprendizaje del algoritmo el cual esta empleado en software libre. Por otra parte, este proceso busca mitigar la clasificación subjetiva por parte del humano el cual se ve influenciado por el nivel de conocimiento de la cultura, permitiendo generar una base en el reconocimiento psicoacústico de géneros folclóricos colombianos, campo el cual no se ha explorado buscando generar una recuperación y restauración de la cultura colombiana planteada en el plan nacional (Uribe, 2002).

5. ALCANCES Y LIMITACIONES

5.1. Alcances

Con el desarrollo del software se clasificó cuatro géneros de las regiones naturales de Colombia determinado por las características de textura tímbrica para un procesamiento y clasificación, así mismo tiene como fortalezas, la utilización de software libre, donde el diseño e implementación del algoritmo soporta la investigación científica en áreas de la ingeniería electrónica y sistemas en la clasificación musical folclórica colombiana.

5.2. Limitaciones

Así mismo la calidad de los audios en parámetros controlados para su procesamiento dependen de la calidad con la que fueron recuperados o grabados en formatos `.wav` donde la única característica de salida del software es el porcentaje de acierto al género del cual pertenece la pieza musical, dicho genero estará delimitado por bambuco, currulao, joropo y vallenato. Otra limitación que se debe tener presente es el hecho que el entrenamiento del algoritmo no debe superar las 15 piezas musicales, debido a que la relación de observaciones de entrenamiento y clasificación debe soportar la diferenciación de los géneros.

6. MARCO TEÓRICO

6.1. Fundamentos teóricos

La forma en que analizamos los datos ha cambiado dramáticamente en los últimos años. Con el advenimiento de las computadoras personales e internet, el gran volumen de datos que tenemos disponible ha crecido enormemente. Las empresas tienen terabytes de datos sobre los consumidores con los que interactúan, y las instituciones de investigación privadas, académicas y gubernamentales tienen una gran cantidad de datos de archivo y encuestas sobre todo tipo de temas de investigación. La información recabada de estos almacenes masivos de datos se ha convertido en una industria en sí misma. Al mismo tiempo, presentar la información de manera fácilmente accesible y digestible se ha vuelto cada vez más desafiante; donde se ha tenido un impacto en cómo se entienden y presentan los resultados, de manera que, una imagen realmente puede valer más que mil palabras, y los seres humanos son muy expertos en extraer información útil de presentaciones visuales. El análisis moderno de datos se basa cada vez más en presentaciones gráficas para descubrir el significado y transmitir resultados. Para resumir, los analistas de datos de hoy en día deben poder acceder a los datos de una amplia gama de fuentes (sistemas de gestión de bases de datos, archivos de texto, paquetes estadísticos y hojas de cálculo), combinar los datos, limpiarlos y anotarlos, analizarlos con el método más reciente, presentan los hallazgos de manera significativa y gráficamente atractiva e incorporan los resultados en informes atractivos que se pueden distribuir a los interesados y al público.

En el presente capítulo se expondrán los conceptos más importantes que se hicieron partícipes en el desarrollo de la investigación.

6.1.1. Muestreo de las señales de audio

Las señales de audio digital se representan fácilmente en entornos de programación por medio de vectores de números reales, como es el caso de cualquier señal de tiempo discreto. Teniendo presente que el término tiempo discreto se refiere al hecho de que, aunque en la naturaleza el tiempo transcurre en un continuo, en el mundo digital solo se pueden manipular muestras de la señal del mundo real que se han extraído en instancias de tiempo discreto. Este proceso se conoce como muestreo y es la primera etapa en la creación de una señal digital de su contra parte del mundo real.

Donde un problema importante en el contexto del muestreo es qué tan alta debe ser la frecuencia de muestreo, el cual, resulta que para muestrear con éxito una señal de tiempo

continuo, la frecuencia de muestreo debe ser igual al menos del doble de la frecuencia máxima de la señal. Este límite inferior de la frecuencia de muestreo se conoce como la tasa de Nyquist (Wikipedia, 2018b) donde se asegura que se evite un fenómeno indeseable conocido como aliasing.

Con esto, en la mayoría de las aplicaciones, la señal de audio se analiza por medio de una técnica de procesamiento llamada de corto plazo, según la cual la señal de audio se divide en ventanas de corto plazo posiblemente superpuestas y el análisis se lleva a cabo. Razón principal por la que se adopta esta técnica de ventanas generalmente es que las señales de audio no son estacionarias por naturaleza, es decir, sus propiedades varían con el tiempo.

Teniendo presente este concepto y con el formalismo matemático y buscando de entender, mediante un ejemplo simplificado, qué significa la no estacionalidad. Más específicamente, considerando una grabación de audio que consiste en una conversación corta (1 s de largo) entre dos personas, la cual es seguida por el grito de una tercera persona (también 1 s de largo). Por lo tanto, esta grabación de audio consta de dos eventos principales: señal de intensidad normal y señal de alta intensidad. Donde la señal cambia bruscamente del estado de la conversación al estado del grito. Desde una perspectiva, esto puede considerarse como un cambio de estacionalidad, es decir, las propiedades de la señal cambian de un estado a otro. En tales situaciones, no tendría realmente sentido calcular, por ejemplo, la intensidad promedio de las muestras de toda la grabación, ya que el valor resultante estaría dominado por las muestras más intensas que se registraron durante el grito. En su lugar, sería más útil dividir la grabación en segmentos cortos y calcular un valor de intensidad por segmento. Esta es también la idea principal detrás del procesamiento a corto plazo. Por supuesto, el cambio de estacionalidad que se observa en el nivel más abstracto de los eventos de audio, mientras que el procesamiento a corto plazo generalmente funciona en el microcosmos de las muestras de las señales.

Teniendo presente, $x(n), n = 0, \dots, N-1$, pertenece a una señal de audio, con N de muestra larga. y en la cual durante el procesamiento a corto plazo, se enfoca en una pequeña parte de la señal. En otras palabras, se sigue un procedimiento de ventanas: en cada paso del procesamiento, donde se multiplica la señal de audio con una versión modificada de una función de ventana de duración finita, $w(n)$, es decir, una señal de tiempo discreto que es cero fuera de una duración finita intervalo y en donde la señal resultante, $x_i(n)$, en el paso i th de procesamiento viene dada por la ecuación:

$$x_i(n) = x(n)w(n - mi), i = 0, \dots, K - 1, \quad (1)$$

donde K es el número de cuadros y m_i es el desfase de desplazamiento, es decir, el número de muestras que se desplaza la ventana para obtener el marco i . La ecuación 1 implica que $x_i(n)$ es cero en todas partes, excepto en la región de las muestras con índices $m_i, \dots, m_i + W_L - 1$, donde W_L es la longitud de la ventana en movimiento. El valor de m_i depende del tamaño del salto, W_S , de la ventana. En donde por ejemplo, si la ventana se desplaza 10 ms en cada salto y la frecuencia de muestreo, F_s , es de 16 kHz, entonces, $m_i = i * W_S * F_s = i * 0,01 * 16000 = i * 160$ muestras, $i = 0, \dots, K - 1$. Además, si $W_L = 300$ muestras, entonces el quinto fotograma comienza en el índice de muestras $160 * 4 = 640$ y termina en el índice de muestras $160 * 4 + 300 - 1 = 939$.

Lo anterior destaca el hecho de que los parámetros importantes en la técnica de procesamiento a corto plazo son la longitud de la ventana, W_L , donde el tamaño de salto, W_S y su tipo. Por lo general, W_L varía de 10 ms a 50 ms. Por otro lado, el paso de la ventana, W_S , controla el grado de superposición entre ventanas sucesivas. Si, por ejemplo, se desea una superposición del 75 % y la longitud de la ventana es de 40 ms, entonces el paso de la ventana debe ser de 10 ms. Se puede deducir fácilmente que el número total, K , de ventanas a corto plazo es igual a:

$$\frac{N - W_L}{W_S} + 1 \quad (2)$$

6.1.2. Transformada de la señal

El objetivo de esta sección es proporcionar una introducción suave a las transformadas de señal seleccionadas, que generan representaciones las cuales son útiles en diversas tareas de audio, incluyendo extracción de características, compresión y análisis de resolución múltiple, por nombrar solo algunas. colocando especial énfasis en descripción de la Transformada Discreta de Fourier.

Transformada Discreta de Fourier La transformada discreta de Fourier (DFT) es de suma importancia en todas las áreas del procesamiento de señales digitales. Se utiliza para derivar una representación en el dominio de la frecuencia de la señal. Por lo tanto, es importante obtener una comprensión firme de la DFT. Con el fin de presentar la transformación desde una perspectiva de implementación y donde el análisis evolucionará en torno a la interpretación de la salida de la DFT y cómo refleja las propiedades de la señal de audio. Antes de continuar, observe que un algoritmo eficiente para el cálculo de los coeficientes de la DFT es el Fast Fourier Transform (FFT), que explota la redundancia computacional en las ecuaciones que definen la DFT y su transformada inversa.

Donde una señal de tiempo discreto, $x(n)$, $n = 0, \dots, N-1$, N muestras largas, su DFT se define como:

$$X(k) = \sum_{n=0}^{N-1} x(n) \exp(j \frac{2\pi}{N} kn), k = 0, \dots, N-1. \quad (3)$$

donde $j \equiv \sqrt{-1}$. Se puede observar que la salida de la transformación es una secuencia de coeficientes N , las $X(k)$ s, que, en general, son números complejos. Donde los coeficientes DFT constituyen la representación en el dominio de la frecuencia de la señal. y en donde la DFT inversa (IDFT) toma como entrada los coeficientes de DFT y devuelve la señal original:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \exp(j \frac{2\pi}{N} kn), n = 0, \dots, N-1. \quad (4)$$

la ecuación 4 proporciona una reconstrucción exacta de la señal original. Como resultado, la señal de dominio de tiempo, $x(n)$, $n = 0, \dots, N-1$ y la señal compleja, $X(k)$, $k = 0, \dots, N-1$ puede ser tratado como representaciones equivalentes. Se puede reescribir la ecuación 4 de la forma:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) y_k(n), n = 0, \dots, N-1. \quad (5)$$

donde $y_k(n) = \exp(j \frac{2\pi}{N} kn)$, $n = 0, \dots, N-1$, entonces se puede ver que la señal original, $x(n)$, se puede escribir como un promedio ponderado de una familia de señales fundamentales, donde cada señal, $y_k(n)$, es un complejo exponencial y su peso es igual al coeficiente de DFT k th.

Una interpretación útil de los coeficientes DFT, en los que se puede estar particularmente interesado por razones prácticas, es que, si F_s es la frecuencia de muestreo que se utilizó para obtener $x(n)$, entonces la exponencial k th corresponde a la frecuencia analógica $f_k = k \frac{F_s}{N}$, $k = 0, \dots, N-1$ (el equivalente en tiempo discreto de los cuales es $\omega k = k \frac{2\pi}{N}$). Para una frecuencia de muestreo dada, los valores más grandes de N dan como resultado un muestreo más denso del eje de frecuencia, porque $\frac{F_s}{N}$ se hace más pequeño. Por lo tanto, se espera que un valor mayor de N produzca una representación más fina de la señal en el dominio de la frecuencia, es decir, una mejor resolución de la frecuencia. Sin embargo, aquí hay un problema sutil, Donde todo análisis sobre el aumento de la resolución de frecuencia

es válida, siempre que la señal permanezca estacionaria, es decir, mientras sus propiedades no cambien con el tiempo. Por ejemplo, una suma de sinusoides de muy larga duración es una señal estacionaria, que se puede estudiar fácilmente con resolución de alta frecuencia. Por otro lado, si una señal cambia de una suma de dos sinusoides a una suma de tres sinusoides, ya no es estacionaria. Solo puede considerarse estacionario a nivel local. En tales situaciones, N no puede aumentar arbitrariamente. Como consecuencia, no siempre se puede lograr la resolución de frecuencia deseable, ya que N está limitada por los cambios de estacionariedad en la señal. En otras palabras, se debe recurrir a un equilibrio entre la resolución de frecuencia y el tiempo.

Transformación De Fourier A Corto Plazo El objetivo de la Transformada de Fourier a corto plazo (STFT) (Coursera, 2018) es dividir la señal en marcos posiblemente superpuestos utilizando una técnica de ventana móvil y calcular la DFT en cada marco. Por lo tanto, la STFT se inscribe en la categoría de técnicas de procesamiento a corto plazo. Teniendo en cuenta, la longitud de la ventana móvil juega un papel importante porque define la resolución de frecuencia del espectro, dada la frecuencia de muestreo. Las ventanas más largas conducen a una mejor resolución de frecuencia a costa de disminuir la calidad de la resolución de tiempo. Por otro lado, las ventanas más cortas proporcionan una representación más detallada en el dominio del tiempo, pero, en general, conducen a una resolución de frecuencia deficiente. En las aplicaciones de análisis de audio, la longitud de la ventana a corto plazo suele oscilar entre 10 y 50 ms. El estándar de audio MPEG7 recomienda que sea un múltiplo de 10 ms.

Si los coeficientes DFT de cada cuadro se colocan en una columna separada de una matriz, la STFT se puede representar como una matriz de coeficientes, donde el índice de la columna representa el tiempo y el índice de la fila está asociado con la frecuencia del coeficiente DFT respectivo. Si se calcula la magnitud de cada coeficiente, la matriz resultante se puede tratar como una imagen y, como resultado, se puede visualizar. Esta imagen se conoce como el espectrograma de la señal y presenta la evolución de la señal en el dominio de frecuencia de tiempo. Para generar el espectrograma, se usa la magnitud o la magnitud cuadrada de los coeficientes de STFT en una escala lineal o logarítmica (dB).

También es importante tener en cuenta que antes de que tenga lugar el cálculo de los coeficientes de DFT, cada cuadro generalmente se multiplica por muestreo con una función de ventana, que apunta a atenuar los puntos finales del cuadro mientras se conserva la señal alrededor del centro del cuadro.

Revisando el tema de la resolución de frecuencia con un ejemplo que gira en torno a la STFT de dos señales moduladas en frecuencia. Antes de continuar con el ejemplo y

recordando sobre la base de un cuadro, la distancia (en Hz) entre dos coeficientes DFT sucesivos es igual a $\frac{F_s}{N}$ Hz, donde F_s es la frecuencia de muestreo y N es el número de muestras del cuadro. La resolución de frecuencia determina cuándo se distinguirán dos frecuencias cercanas en el espectro de la señal. Para una tasa de muestreo fija, F_s , donde mejora la resolución DFT, necesitando aumentar la longitud del marco a corto plazo. Sin embargo, como ya se ha explicado, el precio a pagar es una resolución de tiempo reducida. Teniendo presentes las siguientes dos señales sintéticas:

$$x_1(n) = \cos(2\pi * 500n + 200 \cos(2\pi * n)), \quad (6)$$

$$x_2(n) = \cos(2\pi * 590n + 200 \cos(2\pi * n)), \quad (7)$$

Cada señal es modulada en frecuencia. Considerando, por ejemplo, la primera ecuación: el término $(2\pi * n)$ crea una modulación de señal de 1 Hz, lo que tiene el efecto de que la frecuencia del tono a 500 Hz muestra una variación de 200 Hz. La señal sintética resultante es de 2 s de largo. En este análisis, la frecuencia de muestreo es de 2 kHz. En la etapa final, creamos la suma de las dos señales individuales:

$$x(n) = x_1(n) + x_2(n). \quad (8)$$

6.1.3. Extracción de características

La extracción de características es una etapa importante de análisis de audio. En general, la extracción de características es un paso de procesamiento esencial en las tareas de reconocimiento de patrones y aprendizaje automático. El objetivo es extraer un conjunto de características del conjunto de datos de interés. Estas características deben ser informativas con respecto a las propiedades deseadas de los datos originales. La extracción de características también se puede ver como un procedimiento de reducción de la tasa de datos buscando que el análisis se base en un número relativamente pequeño de características. En algunos casos, los datos originales, es decir, la señal de audio, son voluminosos y, como tales, son difíciles de procesar directamente en cualquier tarea de análisis.

Por lo tanto, es necesario transformar la representación de datos inicial en una más adecuada, extrayendo características de audio que representa las propiedades de las señales originales al tiempo que reduce el volumen de datos. Para lograr este objetivo, es importante tener un buen conocimiento del dominio de la aplicación, para que poder decidir

cuáles son las mejores características. Por ejemplo, al discriminar entre segmentos de voz y música, una característica interesante del candidato es la desviación de la energía de la señal, ya que esta característica tiene un significado físico que encaja bien con la tarea de clasificación en particular.

En la mayoría de los métodos de análisis y procesamiento de audio, la señal se divide primero en cuadros a corto plazo. Este enfoque también se emplea durante la etapa de extracción de características; la señal de audio se divide en cuadros posiblemente superpuestos y se calcula un conjunto de características por cuadro. Este tipo de procesamiento genera una secuencia, F , de vectores de características por señal de audio. La dimensionalidad del vector de características depende de la naturaleza de las características adoptadas. No es infrecuente utilizar características unidimensionales, como la energía de una señal, sin embargo, en la mayoría de las aplicaciones de análisis de audio sofisticadas se extraen y combinan varias características para formar vectores de características de mayor dimensionalidad. La secuencia o secuencias extraídas de los vectores de características se pueden utilizar para el procesamiento / análisis posterior de los datos de audio.

En general, las funciones de audio en el dominio del tiempo se extraen directamente de las muestras de la señal de audio. Ejemplos típicos son la energía a corto plazo y la tasa de cruce por cero a corto plazo. Tales características ofrecen una forma sencilla de analizar señales de audio, aunque generalmente es necesario combinarlas con características de dominio de frecuencia más sofisticadas. Lo que sigue, es la definición de las funciones de dominio de tiempo.

Cruce por cero: La velocidad de cruce por cero (ZCR) de un cuadro de audio es la velocidad de los cambios de signo de la señal durante el cuadro. En otras palabras, es el número de veces que la señal cambia de valor, de positivo a negativo y viceversa, dividida por la longitud del cuadro. La ZCR se define de acuerdo con la siguiente ecuación:

$$Z(i) = \frac{1}{2W_L} \sum_{n=1}^{W_L} |sgn[x_i(n)] - sgn[x_i(n-1)]| \quad (9)$$

donde $sgn(*)$ es la función de signo, es decir

$$sgn[x_i(n)] = (1, x_i(n) \geq 0, -1, x_i(n) < 0). \quad (10)$$

ZCR puede interpretarse como una medida del ruido de una señal. Por ejemplo, usual-

mente exhibe valores más altos en el caso de señales ruidosas. También se sabe que refleja, de una manera bastante aproximada, las características espectrales de una señal. Tales propiedades de la ZCR, junto con el hecho de que es fácil de calcular, han llevado a su adopción por numerosas aplicaciones, incluida la discriminación entre el habla y la música, la detección del habla y la clasificación del género musical. para nombrar sólo unos pocos.

Características De Audio De Dominio De Frecuencia: La Transformada Discreta de Fourier (DFT) de una señal se puede calcular fácilmente con OCTAVE usando la función incorporada. La DFT se usa ampliamente en el análisis de señales de audio porque proporciona una representación conveniente de la distribución del contenido de frecuencia de los sonidos, es decir, del espectro de sonido. Ahora describiendo algunas características de audio ampliamente utilizadas que se basan en la DFT de la señal de audio. Las características de este tipo también se denominan características de audio de dominio de frecuencia.

Donde W_L es el número de muestras por ventana a corto plazo. También es el número de coeficientes DFT del cuadro. En términos generales, para el cálculo de las características espectrales, basta con trabajar con la primera mitad de los coeficientes, porque la segunda mitad sirve principalmente para reconstruir la señal original. Para simplificar la notación, sea W_{fL} el número de coeficientes que se utilizan en los cálculos a seguir.

Centroide espectral: El centroide espectral y la dispersión espectral son dos medidas simples de la forma y la posición espectral. El centroide espectral es el centro de "gravidad" del espectro. El valor del centroide espectral, C_i , del marco de audio i th se define como:

$$C_i = \frac{\sum_{k=1}^{W_{fL}} k X_i(k)}{\sum_{k=1}^{W_{fL}} X_i(k)} \quad (11)$$

La dispersión espectral es el segundo momento central del espectro. Para calcularlo, uno tiene que tomar la desviación del espectro del centroide espectral, de acuerdo con la siguiente ecuación:

$$S_i = \sqrt{\frac{\sum_{k=1}^{W_{fL}} (k - C_i)^2 X_i(k)}{\sum_{k=1}^{W_{fL}} X_i(k)}} \quad (12)$$

El código en OCTAVE que calcula el centroide espectral y la dispersión espectral de una trama de audio se presenta en una función. Teniendo en cuenta que esta función toma como entrada la magnitud de los coeficientes DFT de un cuadro de audio, en lugar del cuadro de audio en sí. En particular, la función de extracción de la estructura, para estudiar cómo se calculan la DFT y las características espectrales relacionadas a corto plazo. Finalmente, normalizamos ambas funciones en el rango $[0,1]$, al dividir sus valores por $\frac{Fs}{2}$. Este tipo de normalización puede ser muy útil cuando las funciones que toman valores de frecuencia se combinan con otras funciones en las tareas de análisis de audio.

Volviendo al centroide espectral, se puede observar que los valores más altos corresponden a los sonidos más brillantes, donde el valor máximo del centroide espectral para segmentos de audio de tres clases de sonidos ambientales. Se puede ver que la clase `.ºtros1`, que consiste principalmente en ruido de fondo, silencio, etc. exhibe valores más bajos para esta estadística, mientras que los valores respectivos son más altos para los sonidos abruptos de las clases `.ºtros2` y `.ºtros3`.

Por otro lado, la dispersión espectral mide cómo se distribuye el espectro alrededor de su centroide. Obviamente, los valores bajos de la dispersión espectral corresponden a señales cuyo espectro se concentra estrechamente alrededor del centroide espectral.

Flujo espectral: El flujo espectral mide el cambio espectral entre dos cuadros sucesivos y se calcula como la diferencia al cuadrado entre las magnitudes normalizadas de los espectros de las dos ventanas sucesivas de corto plazo:

$$Fl_{(i,i-1)} = \sum_{k=1}^{W_{fL}} (EN_i(k) - EN_{i-1}(k))^2, \quad (13)$$

Donde $EN_i(k) = \frac{X_i(k)}{\sum_{l=1}^{W_{fL}} X_i(l)}$ es decir, $EN_i(k)$ es el coeficiente de DFT normalizado kth en la trama i.

Se observa que los valores de flujo espectral son más altos para la clase de habla. Esto se espera considerando que los cambios espectrales locales son más frecuentes en las señales del habla debido a la rápida alternancia entre los fonemas, algunos de los cuales son casi periódicos, mientras que otros son de naturaleza ruidosa.

Atenuacion espectral: Esta característica se define como la frecuencia por debajo de la cual se concentra un cierto porcentaje (generalmente alrededor del 90 %) de la distribución

de magnitud del espectro. Por lo tanto, si el m th coeficiente DFT corresponde a la reducción espectral del marco i , entonces satisface la siguiente ecuación:

$$\sum_{k=1}^m X_i(k) = C \sum_{k=1}^{W_f L} X_i(k) \quad (14)$$

Donde C es el porcentaje adoptado. La frecuencia de reducción espectral generalmente se normaliza al dividirla con $W_f L$, de modo que toma valores entre 0 y 1. Este tipo de normalización implica que un valor de 1 corresponde a la frecuencia máxima de la señal, es decir, a la mitad de la frecuencia de muestreo.

La reducción espectral también se puede tratar como un descriptor de forma espectral de una señal de audio y se puede utilizar para discriminar entre sonidos sonoros y no sonoros. También se puede utilizar para discriminar entre diferentes tipos de pistas de música.

MFCCs: Los Coeficientes de *CepstrumMel – Frequency*(MFCC) han sido muy populares en el campo del procesamiento del habla. Los MFCC son en realidad un tipo de representación cepstral de la señal, donde las bandas de frecuencia se distribuyen de acuerdo con la escala de mel, en lugar del enfoque espaciado linealmente. Para extraer MFCC de un marco, se ejecutan los siguientes pasos:

- Se computa la DFT.
- El espectro resultante se proporciona como entrada a un banco de filtros de escala de mel que consiste en filtros L . Los filtros generalmente tienen respuestas de frecuencia triangular superpuestas. La escala de mel introduce un efecto de distorsión de frecuencia en un intento de cumplir con ciertas observaciones psicoacústicas que han indicado que el sistema auditivo humano puede distinguir las frecuencias vecinas más fácilmente en la región de baja frecuencia. A lo largo de los años se han propuesto varias funciones de distorsión de frecuencia, por ejemplo, la escala mel es una escala de intervalos de frecuencia motivada por la percepción, que, si es juzgada por un oyente humano, se percibe que está espaciada por igual.
- Si está bien, $k = 1, \dots, L$, es la potencia en la salida del filtro k th, luego los MFCC resultantes vienen dados por la ecuación:

$$cm = \sum_{k=1}^L (\log O_k) \cos\left[m\left(k - \frac{1}{2}\right)\frac{\pi}{L}\right], m = 1, \dots, L \quad (15)$$

Por lo tanto, de acuerdo con la ecuación 15, los MFCC son los coeficientes de transformada de coseno discretos del espectro de potencia logarítmica escalada. Donde, los MFCC se han utilizado ampliamente en el reconocimiento de voz, la clasificación del género musical, el agrupamiento de altavoces y muchas otras aplicaciones de análisis de audio.

Los MFCC han demostrado ser características potentes en varias aplicaciones de análisis de audio. Por ejemplo, en la tarea de clasificación binaria del habla frente a la música, exhiben una capacidad discriminativa significativa. De manera que se presenta los histogramas de la desviación estándar de la 2da MFCC para la tarea de clasificación binaria de voz frente a música. Se puede ver que el poder discriminatorio de esta característica es bastante alto, y que el error bayesiano estimado es igual al 11.8% usando esta estadística de característica única. Vale la pena señalar que, según la tarea en cuestión, se han utilizado diferentes subconjuntos de los MFCC a lo largo de los años. Por ejemplo, se ha vuelto habitual en muchas aplicaciones de procesamiento de música seleccionar los primeros 13 MFCC porque se considera que contienen suficiente información discriminativa en el contexto de varias tareas de clasificación.

6.1.4. lenguaje R

En todo el mundo, millones de analistas e investigadores de datos utilizan el lenguaje R para ocuparse de sus problemas más difíciles en los campos que van desde la ciencia computacional hasta el marketing. R Programming, o R (Lantz, 2013), se ha convertido en el lenguaje más frecuente para la ciencia de datos y en una herramienta fundamental para las organizaciones basadas en análisis y finanzas, por ejemplo, Google, Facebook y LinkedIn.

R es un lenguaje extremadamente poderoso para manipular y analizar datos, mediante métodos que son populares en comunidades que desarrollan algoritmos de aprendizaje de máquina, las técnicas estadísticas tienen un alto desarrollo por parte de la comunidad debido a que es un software con licencia GPL, al ser un lenguaje de alto nivel como lo es Python o Ruby, el desarrollo del lenguaje permite realizar procesos matemáticos como lo son la regresión lineal de un modelo con la llamada de una sola línea de código.

La programación en R ofrece una amplia variedad de técnicas estadísticas (modelado directo y no lineal), pruebas estadísticas tradicionales, análisis de la organización del tiempo, agrupación, agrupamiento y técnicas gráficas, y es profundamente extensible. El lenguaje

S es regularmente el vehículo de decisión para la exploración en metodología estadística, y R proporciona una ruta de código abierto para la cooperación en esa acción.

El universo de las aplicaciones de programación abarca desde teorías hipotéticas, estadísticas computacionales hasta ciencias duras, por ejemplo, astronomía, química y genómica hasta aplicaciones prácticas en negocios, avance de medicamentos, finanzas, salud, marketing, medicina y mucho más. Ya que R tiene casi 5,000 paquetes (bibliotecas de funciones), de los cuales una gran parte está comprometida con aplicaciones particulares.

Ahora bien, las bases de datos que encontramos hoy en día en la red buscan modelar un sistema por medio de un análisis tanto estadístico como de la matemática tradicional, que combinado con la ingeniería en software o ciencia computacional permiten el desarrollo de algoritmos de reconocimiento de patrones, esto denominado aprendizaje de máquina. Las herramientas de la estadística tradicional realizan una aproximación al mundo real aunque algunas veces solo permite interpretar cierto grupo de datos, en cambio el aprendizaje de máquina da un entendimiento a los equipos del cómo funcionan ciertos procesos y buscan una optimización de manera lógica en el análisis de grandes cantidades de datos con ayuda de procesos matemáticos básicos y estadísticos tradicionales, con eso el aprendizaje sucede con el entendimiento de los datos mediante las estructuras básicas en algoritmos, encontrando las diferencias entre una señal y ruido, de manera que se realiza un entrenamiento y aprendizaje del grupo de datos, por lo cual la observación y aprendizaje de los mismos permiten reconocer procesos automáticamente, siendo esto la base del aprendizaje de máquina.

Una de las fortalezas de R es la facilidad con la que se pueden producir gráficos de calidad de publicación bien diseñados, incluyendo símbolos matemáticos y fórmulas donde sea necesario. Se ha prestado gran atención a los valores predeterminados para las opciones de diseño menores en gráficos, pero el usuario conserva el control total.

Dejando de lado los conceptos erróneos populares, se obtendrá una comprensión más matizada del aprendizaje automático. Donde se buscara presentar los conceptos fundamentales que definen y diferencian los enfoques de aprendizaje automático más utilizados.

- Los orígenes y aplicaciones prácticas del aprendizaje automático.
- Cómo las computadoras convierten los datos en conocimiento y acción.
- Cómo hacer coincidir un algoritmo de aprendizaje automático con sus datos

El campo del aprendizaje automático proporciona un conjunto de algoritmos que transforman los datos en conocimiento práctico. De manera que usar R para comenzar a aplicar

el aprendizaje automático a problemas del mundo real permite un análisis óptimo de los datos.

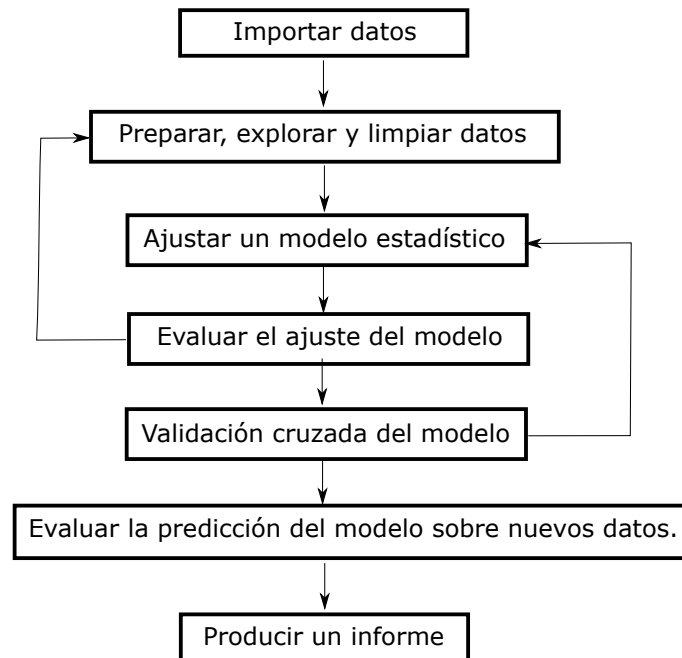


Figura 1: Diagrama procesamiento de datos en R

La tabla 1, 2 es una comparativa de los lenguajes para la ejecución del algoritmo de aprendizaje.

Lenguaje programación	R	Python	MATLAB	Octave
Destacados	Código abierto; Bueno para el análisis estadístico y el procesamiento de datos; enorme colección de algoritmos disponibles como paquetes; soporte de visualización	Código abierto; fácil de aprender; todos los beneficios del lenguaje de programación de propósito general; grandes volúmenes de datos listos	Bueno con procesos matemáticos, operaciones matriciales complejas; Amplio rango de aprendizaje automático;	Código abierto, bueno para cálculos numéricos; Construido con compatibilidad Matlab; conocido como clon de Matlab; Bueno para construir modelos preliminares.
Contratiempos	Curva de aprendizaje empinada; comandos oscuros.	Velocidad de ejecución; necesita manejar las dependencias de la biblioteca si se migran desde 2.x a 3.x	Propietario; Falta de buen ecosistema de código abierto; Dificultad cuando los datos no pueden representarse en matrices	Falta de interoperabilidad con la fuente de datos externa - bases de datos csv, etc
Principales librerías	gbm, RTextTools, dplyr, zoo, ggplot2, caret	Scikit-Learn, pandas, Matplotlib, Numpy, Scipy, theano, nltk	Statistics and machine learning; image processing; Signal processing	libsvm; shogun; liblinear, ltfat; vlfeat

Tabla 1: Tabla lenguajes desarrollo aprendizaje de maquina

Lenguaje programación	R	Python	MATLAB	Octave
Velocidad ejecución	4	4	2	2
Curva aprendizaje	3	4	3	2
Capacidad análisis datos	5	3	4	3
Capacidades gráficas	4	2	4	2
Herramientas	4	2	5	2
Soporte	4	5	3	3
Integración con app externas	3	5	2	2
Entorno laboral	4	5	2	1
Puntaje	31	30	25	17

Tabla 2: Tabla cuantitativa lenguajes desarrollo aprendizaje de maquina

6.1.5. Tipos de algoritmos de aprendizaje automático.

Los algoritmos de aprendizaje automático se dividen en categorías según su finalidad, Figura 2. Comprender las categorías de algoritmos de aprendizaje es un primer paso esencial hacia el uso de datos para impulsar la acción deseada.

Un modelo predictivo se usa para tareas que involucran, como su nombre lo indica, la predicción de un valor utilizando otros valores en el conjunto de datos. El algoritmo de aprendizaje intenta descubrir y modelar la relación entre la característica objetivo (la característica que se predice) y las otras características. A pesar del uso común de la palabra "predicción" para implicar el pronóstico, los modelos predictivos no necesitan necesariamente prever eventos en el futuro. Por ejemplo, un modelo predictivo podría usarse para predecir eventos pasados, como la fecha de la concepción de un bebé utilizando los niveles hormonales actuales de la madre. Los modelos predictivos también se pueden usar en tiempo real para controlar los semáforos durante las horas pico.

Debido a que los modelos predictivos reciben instrucciones claras sobre lo que necesitan aprender y cómo deben aprenderlo, el proceso de entrenamiento de un modelo predictivo se conoce como aprendizaje supervisado. La supervisión no se refiere a la participación humana, sino al hecho de que los valores objetivo proporcionan una manera para que el alumno sepa qué tan bien ha aprendido la tarea deseada. Dicho de manera más formal,

dado un conjunto de datos, un algoritmo de aprendizaje supervisado intenta optimizar una función (el modelo) para encontrar la combinación de valores de características que dan como resultado el resultado objetivo.

El aprendizaje supervisados también se pueden usar para predecir datos numéricos como ingresos, valores de laboratorio, puntajes en pruebas o conteos de artículos. Para predecir tales valores numéricos, una forma común de predicción numérica ajusta los modelos de regresión lineal a los datos de entrada. Aunque los modelos de regresión no son el único tipo de modelos numéricos, son, con mucho, los más utilizados. Los métodos de regresión se usan ampliamente para pronosticar, ya que cuantifican en términos exactos la asociación entre los insumos y el objetivo, incluidos ambos, la magnitud y la incertidumbre de la relación.

Se utiliza un modelo descriptivo para las tareas que se beneficiarían de la información obtenida al resumir los datos de formas nuevas e interesantes. A diferencia de los modelos predictivos que predicen un objetivo de interés, en un modelo descriptivo, ninguna característica individual es más importante que cualquier otra. De hecho, dado que no hay un objetivo que aprender, el proceso de capacitación de un modelo descriptivo se denomina aprendizaje no supervisado. Aunque puede ser más difícil pensar en aplicaciones para modelos descriptivos, después de todo, lo bueno es que un alumno que no está aprendiendo nada en particular, se usa con bastante frecuencia para la extracción de datos.

La tarea de modelado descriptivo de dividir un conjunto de datos en grupos homogéneos se denomina agrupación en clústeres. Esto se usa a veces para el análisis de segmentación que identifica grupos de individuos con un comportamiento similar o información demográfica, por lo que las campañas publicitarias podrían adaptarse a audiencias particulares. Aunque la máquina es capaz de identificar los grupos, se requiere la intervención humana para interpretarlos. Por ejemplo, dados cinco grupos diferentes de compradores en una tienda de comestibles, el equipo de mercadeo deberá comprender las diferencias entre los grupos para crear una promoción que se adapte mejor a cada grupo.

Por último, una clase de algoritmos de aprendizaje automático conocidos como meta-aprendices no está ligada a una tarea de aprendizaje específica, sino que se centra en aprender a aprender de manera más efectiva. Un algoritmo de meta-aprendizaje utiliza el resultado de algunos aprendizajes para informar el aprendizaje adicional. Esto puede ser beneficioso para problemas muy difíciles o cuando el rendimiento de un algoritmo predictivo debe ser lo más preciso posible.

Para comenzar a aplicar el aprendizaje automático a un proyecto del mundo real, se deberá determinar cuál de las cuatro tareas de aprendizaje que representa el proyecto:

clasificación, predicción numérica, detección de patrones o agrupación. La tarea impulsará la elección del algoritmo. Por ejemplo, si está realizando una detección de patrones, es probable que emplee reglas de asociación. De manera similar, un problema de agrupación probablemente utilizará el algoritmo k-means y la predicción numérica utilizará análisis de regresión o árboles de regresión.

Para la clasificación, se necesita más reflexión para hacer coincidir un problema de aprendizaje con un clasificador apropiado. En estos casos, es útil considerar varias distinciones entre algoritmos, distinciones que solo serán evidentes al estudiar cada uno de los clasificadores en profundidad. Por ejemplo, dentro de los problemas de clasificación, los árboles de decisión resultan en modelos que se entienden fácilmente, mientras que los modelos de redes neuronales son notoriamente difíciles de interpretar. Si estaba diseñando un modelo de calificación crediticia, esta podría ser una distinción importante porque la ley a menudo requiere que se notifique al solicitante las razones por las cuales fue rechazado para el préstamo. Incluso si la red neuronal es mejor para predecir los incumplimientos de los préstamos, si no se pueden explicar sus predicciones, entonces es inútil para esta aplicación.

Los principios básicos del aprendizaje automático se trata de crear programas con parámetros ajustables que se adaptan automáticamente para mejorar su comportamiento al adaptarse a los datos vistos anteriormente. El aprendizaje automático puede considerarse un subcampo de la inteligencia artificial, ya que esos algoritmos pueden considerarse bloques de construcción, para hacer que las computadoras aprendan a comportarse de manera más inteligente, generalizando de alguna manera que el almacenamiento y la recuperación de elementos de datos como lo haría un sistema de base de datos.

El proceso de aprendizaje automático, tiene un estándar de aprendizaje y entendimiento a partir de los datos que ingresan a la maquina, los diferentes métodos de aprendizaje presentan características que los permiten entender mediante algoritmos que se encargan de encontrar patrones y le da una clasificación como se observa en la siguiente figura

De acuerdo al estándar hay dos variables categóricas generales, con las cuales se puede realizar un predictor, teniendo presente la cantidad de datos los procesos pueden ser semi-asistido de aprendizaje ó la predicción de los datos mediante etiquetas o proceso asistido de aprendizaje en el caso de que las etiquetas determinen la clasificación: un proceso no asistido como son los clusters donde las etiquetas las determina el sistema mediante un análisis de las distancias de los datos e interpretación de los datos.

(Castro Rey, 2018) Plantea la clasificación como el predecir una variable categórica a partir de un vector aleatorio; partiendo de una variable aleatoria Y , conocida como variable

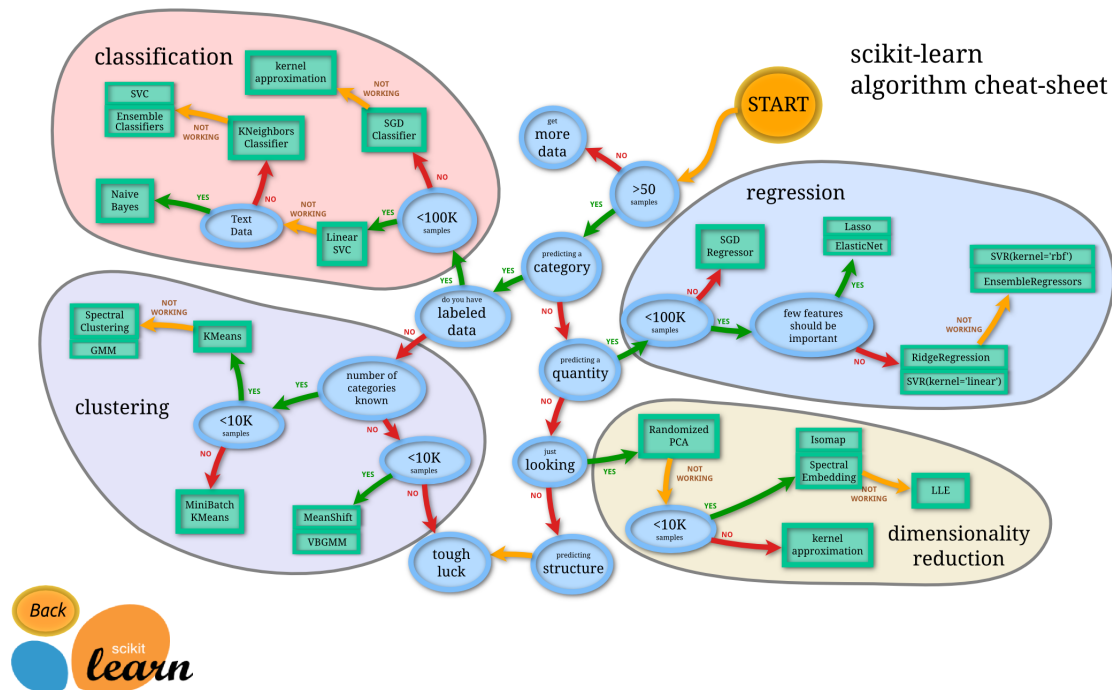


Figura 2: Hoja de trucos del algoritmo scikit-learn Python. tomado:
(Pedregosa y cols., 2011)

objetivo y donde cada elemento del vector corresponde a una clase o categoría diferente. La clasificación de datos se puede entender como un proceso de dos pasos, el primero la fase de entrenamiento o aprendizaje donde se selecciona el subconjunto de datos de entrenamiento; en la segunda etapa de clasificación, donde el algoritmo utiliza el resto de los datos para observar a que clase son asignados y de esta manera poder realizar una evaluación de los errores que comete el algoritmo, de la misma manera se tiene una referencia del error que se cometerá si se incluyen nuevos datos de prueba.

Por otro lado, (Rodrigo, 2017) argumenta el amplio abanico de técnicas no supervisadas cuya finalidad es encontrar patrones; teniendo presente que las observaciones que están dentro del mismo grupo son similares entre ellas y distintas a otros grupos siendo esto un método no supervisado. debido al ignorar la variable respuesta la cual indica a que grupo pertenece es lo que diferencia a las técnicas no supervisadas a las técnicas de clasificación anteriormente explicadas; dada la utilidad del clustering se han desarrollado variedad de algoritmos y métodos los cuales se diferencian en tres grandes grupos.

- Partitioning Clustering.
- Hierarchical Clustering

- Métodos que combinan o modifican los anteriores

Sin embargo, la clasificación por cantidad de datos mediante los métodos de regresión y reducción de dimensionalidad apoyan en la interpretación de los datos y reducción de los mismo de manera que se los algoritmos categóricos realicen reconocimiento de patrones. De manera que los algoritmos de regresión tienen como objetivo el realizar la clasificación e interpretación de los datos, teniendo presente que los datos sean linealmente separables y las variables categóricas tenga cierta relación entre ellas presentando algoritmos como regresión simple, múltiple, gradiente estocástico para datos escasos, Lasso, regresión mediante estimación de densidad, entre otros, siendo estos métodos la partida de análisis inicial en muchos casos.

De acuerdo con esto, con los siguientes algoritmos se plantea realizar un reconocimiento de patrones teniendo en cuenta la cantidad y las características de los datos de ingreso al algoritmo de aprendizaje.

6.1.6. k Vecinos más cercanos

El método de k vecinos más cercanos (en inglés, k -nearest neighbors, abreviado k -NN) (Sirohi, 2018), se definen por su característica de clasificar ejemplos no marcados asignándoles la clase de ejemplos etiquetados similares. A pesar de la simplicidad de esta idea, el método del vecino más cercano es extremadamente poderosos, se han utilizado con éxito para:

- Aplicaciones de visión artificial, incluido el reconocimiento óptico de caracteres y el reconocimiento facial en imágenes fijas y vídeo.
- Predecir si una persona disfrutará de una película o recomendación musical.
- Identificación de patrones en datos genéticos, tal vez para usarlos en la detección de proteínas o enfermedades específicas.

En general, los clasificadores de vecinos más cercanos son adecuados para las tareas de clasificación, donde las relaciones entre las características y las clases objetivo son numerosas, complicadas o extremadamente difíciles de entender, sin embargo, los elementos del tipo de clase similar tienden a ser bastante homogéneos. Otra forma de decirlo sería afirmar que si un concepto es difícil de definir, pero lo sabe cuando lo ve, entonces los vecinos más cercanos podrían ser apropiados. Por otro lado, si los datos son ruidosos y, por lo tanto, no

existe una distinción clara entre los grupos, los algoritmos del vecino más cercano pueden tener dificultades para identificar los límites de la clase.

El algoritmo k -NN obtiene su nombre del hecho de que usa información sobre los vecinos más cercanos a k de un ejemplo para clasificar los ejemplos no etiquetados. La letra k es un término variable que implica que se podría usar cualquier número de vecinos más cercanos. Después de elegir k , el algoritmo requiere un conjunto de datos de entrenamiento compuesto de ejemplos que se han clasificado en varias categorías, como lo indica una variable nominal. Luego, para cada registro no etiquetado en el conjunto de datos de prueba, k -NN identifica k registros en los datos de entrenamiento que son los más cercanos en similitud. A la instancia de prueba sin etiqueta se le asigna la clase de la mayoría de los k vecinos más cercanos. Las fortalezas y debilidades de este algoritmo son las siguientes:

Fortalezas	Debilidades
Simple y eficaz	No produce un modelo, lo que limita la capacidad de comprender cómo se relacionan las características con la clase
No hace suposiciones sobre la distribución de datos subyacente	Fase de clasificación lenta
Fase de entrenamiento rápido	Las características nominales y los datos faltantes requieren un procesamiento adicional

Tabla 3: Tabla Fortalezas y debilidades k -NN

La ubicación de los vecinos más cercanos al dato de entrada requiere una función de distancia o una fórmula que mida la similitud entre los dos casos. Hay muchas maneras diferentes de calcular la distancia. Tradicionalmente, el algoritmo k -NN usa la distancia euclidiana, que es la distancia que se mediría si fuera posible usar una regla para conectar dos puntos.

La distancia euclidiana se especifica mediante la siguiente fórmula, donde p y q son los ejemplos a comparar, cada uno con n características. El término p_1 se refiere al valor de la primera característica del ejemplo p , mientras que q_1 se refiere al valor de la primera característica del ejemplo q :

$$dist(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (16)$$

La decisión de cuántos vecinos usar para k -NN determina qué tan bien se generalizará el

modelo a los datos futuros. El equilibrio entre el ajuste excesivo y el ajuste insuficiente de los datos de capacitación es un problema conocido como compensación de sesgo-variación. La elección de una k grande reduce el impacto o la variación causada por los datos ruidosos, pero puede sesgar al alumno para que corra el riesgo de ignorar patrones pequeños pero importantes.

Suponiendo que se toma la postura extrema de establecer una k muy grande, tan grande como el número total de observaciones en los datos de entrenamiento. Con cada instancia de entrenamiento representada en la votación final, la clase más común siempre tiene la mayoría de los votantes. En consecuencia, el modelo siempre predeciría la clase mayoritaria, independientemente de los vecinos más cercanos.

En el extremo opuesto, el uso de un vecino más cercano permite que los datos ruidosos o los valores atípicos influyan indebidamente en la clasificación de los ejemplos. Por ejemplo, suponiendo que algunos de los ejemplos de entrenamiento fueron etiquetados incorrectamente accidentalmente. Se predeciría que cualquier ejemplo no etiquetado que sea el más cercano al vecino etiquetado incorrectamente tendrá la clase incorrecta, incluso si otros nueve vecinos más cercanos hubieran votado de manera diferente.

La Figura 3 ilustra, de manera más general, cómo el límite de decisión (representado por una línea discontinua) se ve afectado por valores de k mayores o menores. Los valores más pequeños permiten límites de decisión más complejos que se ajustan más cuidadosamente a los datos de entrenamiento. El problema es que no sabemos si el límite recto o el límite curvo representan mejor el verdadero concepto subyacente que se debe aprender.

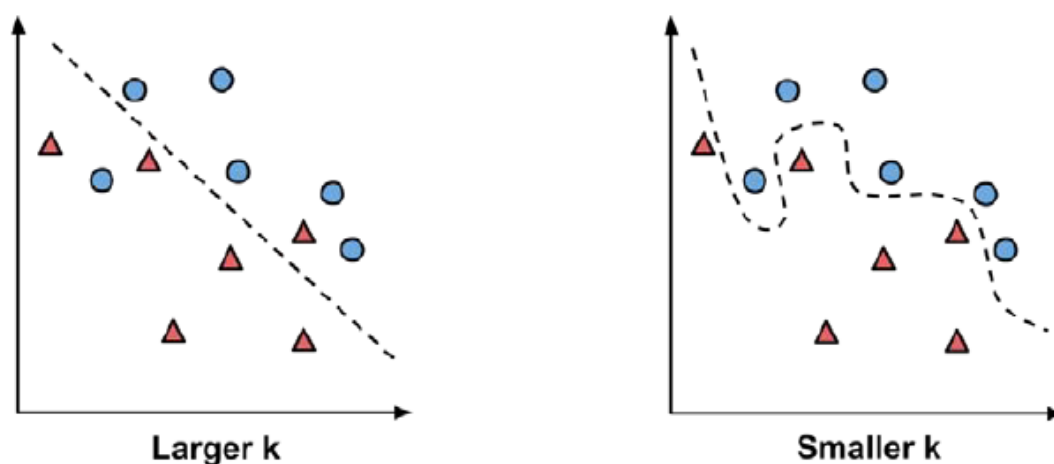


Figura 3: Selección valor k . tomado:
(Lantz, 2013)

En la práctica, elegir k depende de la dificultad del concepto a aprender y del número de registros en los datos de entrenamiento. Una práctica común es comenzar con k igual a la raíz cuadrada del número de ejemplos de entrenamiento. Sin embargo, estas reglas no siempre dan como resultado el mejor k único. Un enfoque alternativo es probar varios valores de k en una variedad de conjuntos de datos de prueba y elegir el que ofrece el mejor rendimiento de clasificación. Dicho esto, a menos que los datos sean muy ruidosos, un gran conjunto de datos de entrenamiento puede hacer que la elección de k sea menos importante. Esto se debe a que incluso los conceptos sutiles tendrán un grupo suficientemente grande de ejemplos para votar como vecinos más cercanos.

6.1.7. Clasificador Bayesiano ingenuo

Los clasificadores basados en métodos bayesianos (Khan, 2017) utilizan datos de entrenamiento para calcular la probabilidad observada de cada resultado en base a la evidencia proporcionada por los valores de características. Cuando el clasificador se aplica más tarde a los datos sin etiquetar, utiliza las probabilidades observadas para predecir la clase más probable para las nuevas características. Es una idea simple, pero resulta en un método que a menudo tiene resultados a la par con algoritmos más sofisticados. De hecho, los clasificadores bayesianos se han utilizado para:

- Clasificación de texto, como el filtrado de correo no deseado (spam).
- Detección de intrusiones o anomalías en redes informáticas.
- Diagnóstico de condiciones médicas dado un conjunto de síntomas observados.

Por lo general, los clasificadores bayesianos se aplican mejor a problemas en los que la información de numerosos atributos debe considerarse simultáneamente para estimar la probabilidad general de un resultado. Mientras que muchos algoritmos de aprendizaje automático ignoran características que tienen efectos débiles, los métodos bayesianos utilizan toda la evidencia disponible para cambiar sutilmente las predicciones. Si un gran número de características tiene efectos relativamente menores, en conjunto, su impacto combinado podría ser bastante grande.

Las relaciones entre los eventos dependientes se pueden describir usando el teorema de Bayes, como se muestra en la fórmula 17. Esta formulación proporciona una forma de

pensar acerca de cómo revisar una estimación de la probabilidad de un evento a la luz de la evidencia proporcionada por otro evento:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (17)$$

La notación $P(A|B)$ se lee como la probabilidad del evento A , dado que el evento B ocurrió. Esto se conoce como probabilidad condicional, ya que la probabilidad de A depende (es decir, condicional) de lo que sucedió con el evento B . El teorema de Bayes nos dice que nuestra estimación de $P(A|B)$ debe basarse en $P(A \cap B)$, una medida de la frecuencia con la que se observa que A y B ocurren juntos, y $P(B)$, una medida de la frecuencia con la que se observa que ocurre B en general.

El teorema de Bayes establece que la mejor estimación de $P(A|B)$ es la proporción de ensayos en los que A ocurrió con B de todos los ensayos en los que ocurrió B . En lenguaje sencillo, esto nos dice que si sabemos que ocurrió el evento B , la probabilidad de que ocurra el evento A es mayor cuanto más a menudo ocurren A y B juntos cada vez que se observa B . En cierto modo, esto ajusta $P(A \cap B)$ para la probabilidad de que ocurra B ; si B es extremadamente raro, $P(B)$ y $P(A \cap B)$ siempre serán pequeños; sin embargo, si A y B casi siempre ocurren juntos, $P(A|B)$ será alto independientemente de la probabilidad de B .

Por definición, $P(A \cap B) = P(A|B) * P(B)$, un hecho que puede derivarse fácilmente aplicando un poco de álgebra a la fórmula anterior. Reorganizando esta fórmula una vez más con el conocimiento de que $P(A \cap B) = P(B \cap A)$ llega a la conclusión de que $P(A \cap B) = P(B|A) * P(A)$, que luego podemos utilizar en la siguiente formulación del teorema de Bayes:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A|B)P(A)}{P(B)} \quad (18)$$

El algoritmo Naive Bayes describe un método simple para aplicar el teorema de Bayes a los problemas de clasificación. Aunque no es el único método de aprendizaje automático que utiliza métodos bayesianos, es el más común. Esto es particularmente cierto para la clasificación de texto, donde se ha convertido en el estándar. Las fortalezas y debilidades de este algoritmo son las siguientes:

Fortalezas	Debilidades
Sencillo, rápido y muy eficaz	Se basa en un supuesto a menudo defectuoso de características igualmente importantes e independientes
Lo hace bien con datos ruidosos y faltantes.	No es ideal para conjuntos de datos con muchas características
Requiere relativamente pocos ejemplos para el entrenamiento, pero también funciona bien con un gran número de ejemplo	Las probabilidades estimadas son menos confiables que las clases predichas

Tabla 4: Tabla Fortalezas y debilidades Clasificador Bayesiano

El clasificador Bayesiano ingenuo se llama así porque hace algunas suposiciones ingenuas sobre los datos. En particular, Naive Bayes asume que todas las características en el conjunto de datos son igualmente importantes e independientes. Estas suposiciones rara vez son ciertas en la mayoría de las aplicaciones del mundo real.

Sin embargo, en la mayoría de los casos en que se violan estas suposiciones, Naive Bayes sigue teniendo un buen desempeño. Esto es cierto incluso en circunstancias extremas donde se encuentran fuertes dependencias entre las características. Debido a la versatilidad y precisión del algoritmo en muchos tipos de condiciones, Naive Bayes es a menudo un fuerte primer candidato para las tareas de aprendizaje de clasificación.

Este algoritmo construye tablas de probabilidades que se utilizan para estimar la probabilidad de que los nuevos ejemplos pertenezcan a varias clases. Las probabilidades se calculan utilizando una fórmula conocida como el teorema de Bayes, que especifica cómo se relacionan los eventos dependientes. Aunque el teorema de Bayes puede ser costoso desde el punto de vista computacional, una versión simplificada que hace suposiciones sobre la independencia de las características es capaz de manejar conjuntos de datos extremadamente grandes.

6.1.8. SVM

Una Máquina de vectores de soporte (SVM) se puede imaginar como una superficie que crea un límite entre puntos de datos trazados en multidimensional que representan ejemplos y sus valores de características. El objetivo de un SVM es crear un límite plano denominado hiperplano, que divide el espacio para crear particiones bastante homogéneas en cada lado. De esta manera, el aprendizaje SVM combina aspectos tanto del aprendizaje del vecino más

cercano basado en la instancia presentado en la seccion 6.1.6 usando vecinos más cercanos, como el modelo de regresión lineal. La combinación es extremadamente poderosa, lo que permite al SVM modelar relaciones altamente complejas.

Aunque las matemáticas básicas que han impulsado el desarrollo (Karatzoglou, Meyer, y Hornik, 2005) SVM han existido durante décadas, recientemente han explotado en popularidad. Por supuesto, esto se basa en su rendimiento de vanguardia, pero quizás también debido al hecho de que los algoritmos SVM galardonados se han implementado en varias bibliotecas populares y bien soportadas en muchos lenguajes de programación, incluyendo R. Por lo tanto, han sido adoptados por una audiencia mucho más amplia, de lo contrario, podrían haber sido incapaces de aplicar las matemáticas algo complejas necesarias para implementar un SVM.

Un algoritmo SVM se pueden adaptar para su uso con casi cualquier tipo de tarea de aprendizaje, incluidas la clasificación y la predicción numérica. Muchos de los éxitos clave del algoritmo han sido en el reconocimiento de patrones. Las aplicaciones notables incluyen:

- Clasificación de los datos de expresión génica de micromatrices en el campo de la bioinformática para identificar cáncer u otras enfermedades genéticas.
- Categorización del texto, como la identificación del idioma utilizado en un documento o la clasificación de documentos por materia.
- La detección de eventos raros pero importantes como fallas en el motor de combustión, brechas de seguridad o terremotos.

SVM utilizan un límite llamado un hiperplano para particionar los datos en grupos de valores de clase similares. Por ejemplo, la siguiente figura muestra los hiperplanos que separan grupos de círculos y cuadrados en dos y tres dimensiones. Debido a que los círculos y los cuadrados se pueden separar perfectamente por la línea recta o la superficie plana, se dice que son linealmente separables.

Los vectores de soporte (indicados por las flechas en la figura que sigue) son los puntos de cada clase que están más cerca del Maximum Margin Hyperplane (MMH); cada clase debe tener al menos un vector de soporte, pero es posible tener más de uno. Usando solo los vectores de soporte, es posible definir el MMH. Esta es una característica clave de SVMs; Los vectores de soporte proporcionan una forma muy compacta de almacenar un modelo de clasificación, incluso si el número de características es extremadamente grande.

Es más fácil entender cómo encontrar el margen máximo bajo el supuesto de que las clases son linealmente separables. En este caso, el MMH está lo más alejado posible de los límites

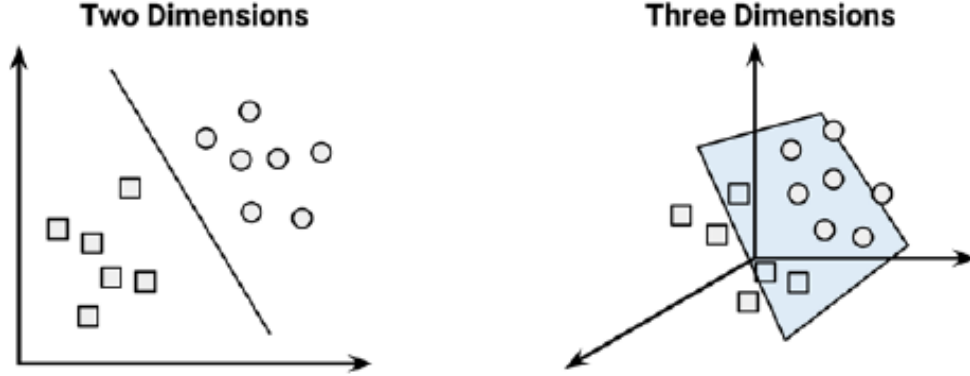


Figura 4: hiperplano SVM. tomado:
(Lantz, 2013)

externos de los dos grupos de puntos de datos. Estos límites exteriores son conocidos como el casco convexo. El MMH es entonces la bisectriz perpendicular de la línea más corta entre los dos cascos convexos. Los sofisticados algoritmos informáticos que utilizan una técnica conocida como optimización cuadrática son capaces de encontrar el margen máximo de esta manera.

Un enfoque alternativo implica una búsqueda a través del espacio de cada posible hiperplano para encontrar un conjunto de dos planos paralelos que dividen los puntos en grupos homogéneos y, al mismo tiempo, están lo más separados posible. Para comprender este proceso de búsqueda, tendremos que definir exactamente lo que queremos decir con un hiperplano. En el espacio n-dimensional, se utiliza la siguiente ecuación:

$$\vec{w} * \vec{x} + b = 0 \quad (19)$$

Usando esta fórmula, el objetivo del proceso es encontrar un conjunto de ponderaciones que especifiquen dos hiperplanos, de la siguiente manera:

$$\vec{w} * \vec{x} + b \geq 1 \quad \vec{w} * \vec{x} + b \leq -1 \quad (20)$$

También requeriremos que estos hiperplanos se especifiquen de tal manera que todos los puntos de una clase caigan por encima del primer hiperplano y todos los puntos de la otra

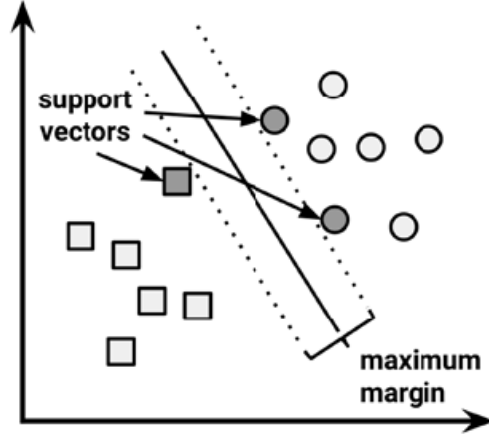


Figura 5: Maximum Margin Hyperplane. tomado:
(Lantz, 2013)

clase queden por debajo del segundo hiperplano. Esto es posible siempre que los datos sean linealmente separables.

La geometría vectorial define la distancia entre estos dos planos como:

$$\frac{2}{\|\vec{w}\|} \quad (21)$$

Aquí, $\|w\|$ indica la norma euclidiana. Porque $\|w\|$ está en el denominador, para maximizar la distancia, necesitamos minimizar $\|w\|$. La tarea normalmente se expresa como un conjunto de restricciones, de la siguiente manera:

$$\min \frac{1}{2} (\|\vec{w}\|)^2 \quad (22)$$

$$\text{s.t. } y_i (\vec{w} \cdot \vec{x}_i - b) \geq 1, \forall \vec{x}_i \quad (23)$$

Al igual que con el otro método para encontrar el margen máximo, encontrar una solución a este problema es una tarea que queda mejor para el software de optimización cuadrática.

Aunque puede ser intensivo en el procesador, los algoritmos especializados son capaces de resolver estos problemas rápidamente incluso en conjuntos de datos bastante grandes.

A medida que se analiza la teoría detrás de los SVM surge la incertidumbre de ¿qué sucede si los datos no son linealmente separables? de manera que la solución a este problema es el uso de una variable de holgura, que crea un margen suave que permite que algunos puntos caigan en el lado incorrecto del margen; donde, se aplica un valor de costo (indicado como C) a todos los puntos que violan las restricciones, y en lugar de encontrar el margen máximo, el algoritmo intenta minimizar el costo total. Por lo tanto, podemos revisar el problema de optimización para:

$$\min \frac{1}{2}(|\vec{w}|)^2 + C \sum_{i=1}^n \xi_i \quad (24)$$

$$\text{s.t. } y_i(\vec{w} * \vec{x} - b) \geq 1 - \xi_i, \forall \vec{x}_i, \xi_i \geq 0 \quad (25)$$

La pieza importante a comprender es la adición del parámetro de costo C , la modificación de este valor ajustará la penalización, por ejemplo, la caída en el lado equivocado del hiperplano. Cuanto mayor sea el parámetro de costo, más difícil será la optimización que intentará lograr una separación del 100 por ciento. Por otro lado, un parámetro de menor costo colocará el énfasis en un margen general más amplio. Es importante lograr un equilibrio entre estos dos para crear un modelo que generalice bien los datos futuros.

En muchas aplicaciones del mundo real, las relaciones entre variables son no lineales, una SVM aún puede capacitarse en dichos datos mediante la adición de una variable de holgura, lo que permite que algunos ejemplos se clasifiquen erróneamente. Sin embargo, esta no es la única manera de abordar el problema de la no linealidad. Una característica clave de los SVM es su capacidad para mapear el problema en un espacio de dimensión superior usando un proceso conocido como el truco del núcleo. Al hacerlo, una relación no lineal puede parecer repentinamente bastante lineal.

Las SVM con núcleos no lineales agregan dimensiones adicionales a los datos para crear una separación de esta manera. Esencialmente, el truco del núcleo implica un proceso de construcción de nuevas características que expresan relaciones matemáticas entre características medidas; esto permite a SVM aprender conceptos que no se midieron explícitamente

en los datos originales. Los SVM con núcleos no lineales son clasificadores extremadamente potentes, aunque sí tienen algunas desventajas, como se muestra en la siguiente tabla:

Fortalezas	Debilidades
Se puede utilizar para problemas de clasificación o predicción numérica.	Encontrar el mejor modelo requiere probar varias combinaciones de núcleos y parámetros de modelo
No está demasiado influido por datos ruidosos y no es muy propenso a sobreajustarse	Puede ser lento para entrenar, especialmente si el conjunto de datos de entrada tiene una gran cantidad de funciones o ejemplos
Puede ser más fácil de usar que las redes neuronales, particularmente debido a la existencia de varios algoritmos SVM bien soportados	Resulta en un modelo complejo de caja negra que es difícil, si no imposible, de interpretar

Tabla 5: Tabla Fortalezas y debilidades SVM

Las funciones del núcleo, en general, son de la siguiente forma. La función denotada por la letra griega phi, es decir, $\phi(x)$, es un mapeo de los datos en otro espacio. Por lo tanto, la función general del núcleo aplica alguna transformación a los vectores de características x_i y x_j los combina con el producto de puntos, que toma dos vectores y devuelve un solo número.

$$k(\vec{x}_i, \vec{x}_j) = \phi\vec{x}_i * \phi\vec{x}_j \quad (26)$$

Usando esta forma, las funciones del kernel han sido desarrolladas para muchos dominios diferentes de datos. Algunas de las funciones del kernel más utilizadas se enumeran a continuación:

El kernel lineal no transforma los datos en absoluto. Por lo tanto, se puede expresar simplemente como el producto puntual de las características:

$$k(\vec{x}_i, \vec{x}_j) = \vec{x}_i * \vec{x}_j \quad (27)$$

El núcleo polinomial de grado d agrega una transformación simple no lineal de los datos:

$$k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i * \vec{x}_j + 1)^d \quad (28)$$

El kernel sigmoide da como resultado un modelo SVM algo análogo a una red neuronal que utiliza una función de activación sigmoide. Las letras griegas kappa y delta se utilizan como parámetros del núcleo:

$$k(\vec{x}_i, \vec{x}_j) = \tanh(\kappa \vec{x}_i * \vec{x}_j - \delta) \quad (29)$$

El kernel RBF gaussiano es similar a una red neuronal RBF. El kernel RBF se desempeña bien en muchos tipos de datos y se piensa que es un punto de partida razonable para muchas tareas de aprendizaje:

$$k(\vec{x}_i, \vec{x}_j) = e^{-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2}} \quad (30)$$

No hay una regla para hacer coincidir un kernel con una tarea de aprendizaje en particular. El ajuste depende en gran medida del concepto a aprender, así como de la cantidad de datos de entrenamiento y las relaciones entre las características. A menudo, se requiere un poco de prueba y error al capacitar y evaluar varias SVM en un conjunto de datos de validación. Dicho esto, en muchos casos, la elección del kernel es arbitraria, ya que el rendimiento puede variar ligeramente.

6.1.9. Folclor Colombiano

El plan nacional de música para la convivencia define la musica como un lenguaje que a partir de elementos de base como las palabras y sus gramáticas se pueden establecer tantas maneras diferentes de hablar como individuos hay que hablen una misma lengua, ya sea por los acentos que emplean para pronunciar esas palabras, o por el significado que les atribuyen. De igual forma, partiendo de los sonidos que se arrancan al cuero de un tambor cununo pueden construirse bases rítmicas y tímbricas únicas que varían entre los miembros de una misma familia del pacífico nariñense.

Teniendo presente que en el marco del Plan Nacional de Desarrollo (Uribe, 2002) se puso en marcha el PNMC como política pública cultural prioritaria para el Estado se propusieron

11 “ejes musicales” definidos a partir de las prácticas más reconocidas de cada región. Estos 11 ejes que dan cuenta de contextos, usos y funciones sociales, elementos sonoros y musicales, realidades simbólicas de las prácticas musicales, son los siguientes:

- Músicas Isleñas (Calypso, Schottis y otros): San Andrés y Providencia.
- Músicas de Acordeón y Cuerdas (Vallenato y otros): Guajira, Cesar y Magdalena.
- Músicas de Pitos y Tambores (De Gaitas Larga y Corta, Millo, Baile Canta’o, Tambora y Bandas tipo pelayera): Atlántico, Bolívar, Sucre y Córdoba.
- Chirimías y Cantos Tradicionales (porro Chocoano, Alabaos y otros): Chocó.
- Músicas de Marimba y Cantos Tradicionales (Currulao y otros): Litoral Pacífico del Valle, Cauca y Nariño.
- Músicas Andinas Sur—Occidente (Bandas de flautas, Vals y otros): Cauca, Nariño y occidente del Putumayo.
- Músicas Andinas Centro-Sur (Rajaleña, Caña, Sanjuanero y otros): Huila y Tolima.
- Músicas Andinas Centro-Oriente (Rumba, Bambuco, Guabina y otros): Norte de Santander, Santander, Boyacá y Cundinamarca.
- Músicas Andinas Centro-Occidente (Pasillo, Bambuco, Shotis y otros): Valle, Antioquia, Quindío, Risaralda y Caldas.
- Músicas Llaneras (Joropo): Vichada, Arauca, Guaviare, Meta, Casanare y Oriente de Cundinamarca y Boyacá.
- Músicas de Cuerdas, Murgas y otras: Caquetá, Amazonas, Putumayo, Guaviare, Vaupés y Guainía.

Música isleña En el territorio de San Andrés, Providencia y Santa Catalina se hace presente una variedad muy amplia de formas musicales que son interpretadas por buena parte de la población, pero su principal característica es que conviven a diario combinándose entre ellas el calypso y el mento con canciones y bailes de salón europeos que tienen más de quinientos años de antigüedad.

Estas formas musicales, que constituyen la base de un sonido tradicional, provienen de distintos lugares y culturas y se han convertido en la identidad musical de las islas en las que además se combinan lenguas como el inglés, el creole y el español. Entre ellas también

se encuentran otras formas típicas como el foxtrot, el waltz, el pasillo, la mazurca, la polca y los shottish, que se unen y suenan junto a la soca, el suck y el reggae, que son más populares y no tan antiguas.

Principales géneros:

- | | | |
|---------------|------------|-----------|
| ■ Calypso | ■ Shottish | ■ Mento |
| ■ Reggae | ■ Foxtrot | ■ Waltz |
| ■ Mazurca | ■ Polca | ■ Pasillo |
| ■ Soca y Suck | | |

Instrumentos:

- | | | |
|-----------------------|-------------|---------------------|
| ■ Carraca o mandíbula | ■ Mandolina | ■ Tináfono o tinajo |
| ■ Guitarra | ■ Violín | ■ Acordeón |

Música vallenata Actualmente se han identificado cinco focos de influencia que son los que determinan las escuelas de música tradicional para las músicas vallenatas. El primero de ellos es Valledupar, que domina la Zona central, el segundo es El Paso, que funciona como epicentro de la zona negroide. Continúan la Zona Ribana, cuyo foco es Fonseca, y una Zona Ribereña, de la que Plato es su epicentro. Cierra el mapa la zona de influencia del vallenato en guitarra que tiene como eje a Codazzi en el Cesar.

La música que interpretaban pasó a popularizarse en la primera mitad del siglo XX como música vallenata, pero sus inicios sonoros ya estaban arraigados en la cultura tradicional del litoral. El acordeón sustituyó instrumentos como las gaitas y las flautas de millo y se alejó de los orígenes para constituirse como insignia de un género musical independiente.

Principales géneros:

- | | |
|--------|------------|
| ■ Son | ■ Paseo |
| ■ Puya | ■ Merengue |

Instrumentos:

- | | |
|------------------|------------------|
| ■ Acordeón | ■ Caja vallenata |
| ■ Guacharaca | ■ Guitarra |
| ■ Bajo eléctrico | |

Músicas de Pitos y Tambores Cimarrones e indígenas entraron a participar en un proceso de mestizaje en el que se fundieron las tradiciones de los blancos, negros e indios. A lo largo del litoral se difundieron gaitas, ‘pitos’ o flautas y tambores, a los que se sumaron luego, durante el siglo XIX, instrumentos como los platillos, el bombardino, las trompetas y los clarinetes con los que harían su intervención las bandas que llegaron para interpretar bailes de salón como mazurcas, polcas y vales, que hoy interpretan los ritmos tradicionales y representan un fuerte movimiento de bandas pelayeras en las sabanas.

La cumbia, el porro y la puya, entre otras formas musicales de la zona, fueron instrumentales en un principio, pero el canto no tardó en aparecer.

Principales géneros:

- | | | |
|------------|---------------|----------|
| ■ Cumbia | ■ Bullerengue | ■ Mapalé |
| ■ Gaita | ■ Puya | ■ Chandé |
| ■ Chalupa | ■ Guacherna | ■ Porro |
| ■ Fandango | | |

Instrumentos:

- | | | | |
|---------------|-------------|----------------|-----------------|
| ■ Gaitas | ■ Pitos | ■ Arco musical | ■ Cañaemillo |
| ■ Guacharaca | ■ Guache | ■ Tablitas | ■ Bombos |
| ■ Redoblantes | ■ Platillos | ■ Campanas | ■ Tambor alegre |
| ■ Tambora | ■ Llamador | | |

Músicas del pacífico Musicalmente el Pacífico colombiano se divide en dos zonas bien diferenciadas: La primera es el norte, que está determinada por el departamento del Chocó y en la que la chirimía, una manifestación musical heredera de las bandas militares tradicionales que llegaron con los españoles al continente, convive con los alabaos, arrullos y cantos de boga que se hacen sentir entre los ríos.

En el sur, el territorio conformado por la zona costera de los departamentos de Cauca, Valle del Cauca y Nariño, la principal expresión musical está marcada por la marimba, un instrumento fabricado con láminas de madera de chonta y resonadores de bambú, que al ser interpretado abre paso a una actividad fundamental en las tradiciones del litoral: el

currulao. Los cuerpos de una pareja establecen un diálogo sin tocarse. El hombre corteja a la mujer, quien en un principio no le presta atención y lentamente va cediendo a las insinuaciones del parejo. Cuando éste obtiene respuesta demuestra su hombría castigando el piso con sus pies descalzos. La marimba suena todo el tiempo y otras parejas se suman al baile, cargando la atmósfera con un rito en el que la tensión entre hombres y mujeres es febril.

Principales géneros:

- | | | | |
|------------------|---------------|------------|--------------------|
| ■ Cantos de boga | ■ Gualí | ■ Chigualo | ■ Tamborito |
| ■ Mazurca | ■ Contradanza | ■ Jota | ■ Abozao |
| ■ Aguabajo | ■ Pasillo | ■ Bambazú | ■ Porro chocoano |
| ■ Currulao | ■ Joga | ■ Patacoré | ■ Bambuco viejo |
| ■ Pango | ■ Caderona | ■ Berejú | ■ Velorio de santo |
| ■ Novenario | ■ Chigualo | | |

Instrumentos:

- | | | |
|---------------------------|--------------------------|--------------|
| ■ Flauta traversa | ■ Clarinete | ■ Bombardino |
| ■ Acordeón | ■ Tambora | ■ Redoblante |
| ■ Platillos | ■ Marimba | ■ Guasá |
| ■ Cununo (hembra y macho) | ■ Bombo (hembra y macho) | |

Música andina centro La música la región andina es música campesina que rompe la idiosincrasia de hombres y mujeres enjutos, para permitirse cantarle a las situaciones cotidianas de las que resulta imposible hablar o quejarse, porque ocupan la totalidad de la existencia. Es música, además, que se revitaliza día a día, pues ha recibido la fuerza de jóvenes que la reinventan, inyectando a la tradición el sentir de una academia que vuelve a las raíces y abre paso a vanguardias y nuevas sonoridades.

Principales géneros:

- Pasillo
- Guabina
- Bambuco caucano
- Torbellino
- Rumba criolla
- Danza
- Bambuco
- Sanjuanero

Instrumentos:

- Quiribillo
- Tambora
- Guitarra
- Bandola andina
- Esterilla
- Flautas de caña
- Tiple
- Chucho
- Riolina o dulzaina
- Requinto

Música andina sur La música tradicional producida por los campesinos del Huila y el Tolima le canta al goce permanente de estar vivo. Las coplas, utilizadas dentro del rajaleña de una manera mordaz y graciosa, le han permitido a los habitantes de la zona expresar los diferentes momentos de la vida al aire libre valiéndose de personajes que todos conocen y admiran, o de los que se burlan cariñosamente. A su manera, los pobladores del Macizo Colombiano y su área de influencia en Cauca, Nariño y Putumayo, han encontrado en las flautas y otros vientos, en los instrumentos de Una cordillera que se moviliza con la música cuerdas y en la percusión, la forma de construir sus sonidos para dialogar con un paisaje de cordilleras anudadas, por donde los Andes entran a Colombia, se despliegan perfilando rostros y territorios.

Principales géneros:

- Rajaleña
- Guabina
- Sanjuanero
- Bambuco
- Rumba criolla
- Son sureño
- Bambuco fiestero
- Danza
- Pasillo
- Caña
- Bambuco viejo
- Huayno

Instrumentos:

- Carraca
- Maracas
- Bandola
- Chucho
- Mates
- Quena
- Esterilla
- Tambora
- andina
- Redoblante
- Flautas
- Tiple
- Quiribillo
- Guache
- Requinto
- Triángulo
- traversas
- Marrana
- Guitarra
- Carángano
- Bombo
- Zampoña

Música Llanera Cuando las lluvias se esconden y le ceden paso a las brutales sequías bajo las que el territorio de los Llanos Orientales se ha forjado, el hombre llanero organiza un joropo para alumbrar a un santo. Hay bebida, baile, música de arpa y maraca, carne en abundancia y familia. Se le pide al santo que envíe lluvias, y se bebe y se baila en su nombre. Todos los parientes, trabajadores y llaneros del ható que organiza el parrando se hacen presentes, y el joropo, animado incansablemente por el arpa y las maracas se puede extender tres días con sus noches.

Principales géneros:

- | | | | |
|--------------|--------------|-------------------|-------------|
| ■ Carnaval | ■ San Rafael | ■ Quirpa | ■ Periquera |
| ■ Chipola | ■ Pajarillo | ■ Zumba que zumba | ■ Gaván |
| ■ Guacharaca | ■ Galerón | ■ Perro de agua | |

Instrumentos:

- | | | | |
|-----------|-------------------|------------------|-------------|
| ■ Arpa | ■ Bandola llanera | ■ Bandolín | ■ Guitarro |
| ■ Cuatro | ■ Maracas | ■ Bajo eléctrico | ■ Sirrampla |
| ■ Furruco | | | |

Música de frontera Las músicas tradicionales de frontera del territorio colombiano se alimentan de la variedad lingüística y racial, de nuevos ritmos que se replican desde puertos como Manaos y Leticia en la cuenca amazónica, de las zonas que limitan con Venezuela, tanto en el Norte de Santander como en los Llanos Orientales y en La Guajira, del Tapón del Darién en el que Panamá y Colombia se funden, de lugares como Puerto Leguízamo, Mitú, Inírida, Iquitos o Tulcán en Ecuador. Su riqueza y supervivencia dependen de las idas y vueltas por fronteras en las que los límites territoriales se entrecruzan por la música.

Principales géneros:

- | | | | |
|------------|-------------------|-----------|----------------|
| ■ Samba | ■ Marcha | ■ Baiões | ■ Forró |
| ■ Dobrado | ■ Xote (chotises) | ■ Batuque | ■ Vals criollo |
| ■ Mixtiana | ■ Marinera | ■ Hwayno | ■ Bambuco |
| ■ Pasillo | ■ Joropo | ■ Lambada | ■ Carimbó |
| ■ Porro | ■ Porrosamba | | |

6.2. ESTADO DEL ARTE

6.2.1. Clasificación a nivel mundial

Este trabajo incorpora elementos a través de los cuales se pretende priorizar la observación, análisis e interpretación de la clasificación de la música normalmente relacionadas como la instrumentación o textura tímbrica, el contenido rítmico y el contenido armónico o tonal de la misma; a nivel mundial, durante el transcurso de los años teniendo como uno de los grandes guías en el campo de la clasificación digital musical al señor George Tzanetakis (Tzanetakis, Essl, y Cook, 2002), el cual plantea una clasificación musical automática de géneros musicales por medio de señales de audio; por medio de unas características especiales, que pueden ser detectadas mediante un análisis detenido de los diferentes archivos de audio, donde el autor plantea que las características son relacionadas con las textura, el timbre, y la instrumentación; se denotan las características superficiales de la música las cuales incluyen la media centroide, la media roll off, la media flux, la media de cruce por cero, y los niveles bajos de energía. El autor además plantea unas características determinadas desde el ritmo, las cuales son determinadas por la transformada de wavelet, en las cuales ubica una serie de bancos de filtros para la caracterización de los audios determinando, una rectificación de la onda, un muestreo descendente, normalizando las características. Con la extracción de estas características el autor plantea una clasificación de diferentes ritmos ubicados en la web y en emisoras de radio, usando una clasificación gaussiana representada en una multidimensional sencilla, distribuyendo las características extraídas para la clasificación por ritmos, donde se combinan las características para la clasificación por géneros, y para diferenciar el tipo de interprete (hombre, mujer); usando el coeficiente medio de frecuencia central (MFCC).

Por otro lado, se presentan nuevas formas de clasificación mediante el uso del espectro del beat (Foote y Uchihashi, 2001) en las cuales se plantea que el ser humano que alguna vez haya tocado un instrumento sabe diferenciar los beats por segundo en una canción en el tiempo a la música con el cual se puede hacer un análisis de ritmo. Aunque es simple para los humanos, esta tarea es considerablemente más difícil de automatizar. Para ellos se introduce una nueva medida del análisis del tempo llamado el espectro del golpe. Esta es una medida de la auto estimación acústica versus al tiempo de retardo, calculado a partir de una representación de la similitud espectral. Los picos en el espectro del golpe corresponden a los principales componentes rítmicos del audio detectados mediante el cruce por cero. El tiempo de repetición de cada componente puede ser determinado por el retardo de cadencia de cada beat, mientras que las amplitudes relativas de los diferentes picos reflejan las fortalezas de sus componentes rítmicos. También presentan el espectrograma beat que

ilustra gráficamente la variación rítmica en el transcurso del tiempo. El latido en el espectrograma es una imagen formada a partir del espectro de latidos en sucesivas ventanas. Los autores plantean el uso de un algoritmo el cual realiza el cálculo del espectro del beat de un audio usando tres principales pasos. Primero, parametrizando el audio mediante el espectro y representando en un vector de características. Segundo, tomando la medida de distancia que se utiliza para calcular la similitud entre las características de pares en el vector para la secuencia de audio. Tercero, la representación de estas características en una matriz de similitud la cual está en dos dimensiones, obteniendo el espectro para los periodos de la señal, usando la suma de la diagonal de la matriz o auto correlacionando los datos.

Por otra parte, (Gouyon, 2006) incorpora elementos a través de los cuales se pretende priorizar la observación, análisis e interpretación de la clasificación de la música, normalmente relacionando la melodía, el ritmo y el contenido armónico o tonal como una unidad en una pieza musical y es donde diserta que cada una de estas características brindan un elemento especial a una composición musical, de esta manera, el ritmo como elemento principal del contenido de una canción, presentando este término como un elemento que el humano tiene inherente en su ser en el cual una persona sin formación musical tiene la capacidad de seguir el ritmo en acciones simples como silbar o tararear, de esta manera Joseph Jordania sugiere que el ritmo se desarrolló en las primeras etapas de la evolución de los homínidos, disertando que numerosos animales caminan rítmicamente y escuchan los sonidos de los latidos maternos; donde todos hacen uso de la capacidad del ritmo para unir a los humanos en una identidad colectiva compartida. Por otro lado, los sistemas computacionales son incapaces por si solos de distinguir el ritmo en una pieza musical, es por esto que se hace uso de software o hardware que le permite a la computadora generar y distinguir las características en una pieza musical; de esta manera, plantea abordar la representación rítmica como elemento central en el reconocimiento rítmico en sistemas computacionales, buscando describir y evaluar diferentes estrategias para calcular funciones periódicas rítmicas y demuestra la utilidad de la funcionalidad de los sistemas al percibir el tempo y ritmos estimando la similitud rítmica o reconociendo diversos géneros musicales.

Además, con el análisis de estructura temporal de datos musicales para clasificación (López García, 2014), argumenta el no haber encontrado registro histórico que compruebe la carencia o falta de medios culturales que manifiesten danza o música, por lo que este medios no solo complementan la cultura, sino la determinan de modo particular; buscando de esta manera definir de una forma sencilla las características que no requieren de ser reproducidas para ser clasificadas, por personas con un oído entrenado, permitiendo la facilidad en la caracterización de los ritmos, por instrumentación y otras características

que permiten al humano realizar una clasificación adecuada. Sin embargo, estas mismas características se puede programar de manera adecuada en la máquina para que logre la clasificación; todo este proceso de análisis de la estructura se encuentra soportada en las características a corto plazo, características a medio plazo, características a largo plazo. Buscando con esta caracterización de la información el sistematizar la extracción de la información necesaria para las características Coeficientes Cepstrales en Escala Mel (MFCC) y la Representación Basada en Armónicos (HR), haciendo uso de una ventana de tiempo mayor el análisis de las diferentes características se torna de una manera menos compleja, permitiendo generar una clasificación mucho más sencillo.

A su vez, la clasificación musical también se ha visto por (dos Reis, 2014) para una aproximación genética a la transcripción automática de música, teniendo en cuenta que la extracción de estas características ha sido un proceso humano en el proceso de notación musical de una señal acústica, teniendo como precedente que la transcripción de la música es un proceso que un software está en la capacidad de realizar mediante la configuración adecuada haciendo uso de modelos no semánticos en formato PCM, para la transcripción automática. Ejerciendo uso de una investigación sobre el procesamiento digital de señales, el aprendizaje de la maquina e indagando sobre el análisis psicomusical para la percepción de una teoría musical. La transcripción del sistema tiene la ayuda de un escritor musical el cual evalúa las grabaciones de audio y determina ciertas características que le permiten al sistema una transcripción automática mediante la identificación de artistas, diferentes géneros y algunos otros componentes que asisten al sistema. Además, el autor hizo uso del algoritmo genético para la solución del problema en la estimación de los múltiples-tonos, para iniciar con una sintetización simple de los modelos de instrumentos musicales, para el experimento además se incluyeron diferentes características con las cuales se logró la transcripción de los diferentes ritmos, con los cuales tuvo inconvenientes con las armonías en las cuales los diferentes timbres se diferencian en un prospecto y modelo espectral envolvente, modelado para la caracterización.

Con todo el proceso de globalización y digitalización de los diferentes campos en el análisis de señales de audio, han surgido múltiples modos de clasificación digital de música permitiendo el surgimiento de nuevas tecnologías que permiten una optimización de estos procesos, tal cual como lo sustenta (Igor y Subramanian, 2002), donde hace uso de la clasificación de Markov para géneros musicales, donde hace uso de diferentes métodos en los cuales y técnicas las cuales con software que realizan clasificación digital de música. Además, se explica que el modelo de Markov diferencia las características, eliminando ciertos problemas que genera la caracterización. El modelo secuencial del proceso realiza un proceso visible para las salidas del algoritmo y un trabajo invisible para los estados

internos, permitiendo usar este método en la aplicación de la detección de los diferentes parámetros necesarios para la clasificación. Por otro lado, algunos tipos de música han demostrado ser más difíciles de clasificar que otros. En particular, el conjunto de datos celtas ha demostrado ser difícil de distinguir de los clásicos y categorías del Rock. Donde en la música particularmente escogida, la instrumentación clásica es a menudo utilizada y ritmo rápido, tambor y base rítmica, donde la música de baile celta es a menudo similar al Rock. Además, el techno ha demostrado ser más fácil de clasificar. Esto tiene sentido intuitivo porque techno y el trance son los géneros totalmente diferentes.

(Chowning y cols., 1984) plantea un Sistema inteligente para el análisis de señales acústicas digitalizadas, el cual busca el realizar una abstracción de características para digitalizar señales acústicas combinadas con técnicas de análisis basadas en el conocimiento de programación por métodos y la detección de características de bajo nivel, con programación con métodos heurísticos, los cuales permitan aprovechar la extracción de los diferentes parámetros, donde la variedad de técnicas usadas en el pre-procesamiento de las señales de audio, mediante el análisis estándar de la transformada de Fourier, permite formular diferentes procesos algorítmicos que permites diferenciar los diferentes tipos de instrumentos musicales. Donde el autor busco combinar las diferentes técnicas de digitalización de las señales de audio, aplicando el uso de algoritmos que permitiera extraer diferentes parámetros los cuales son usados para clasificación digital, mediante el uso de las diferentes herramientas de análisis del espectro de una señal realizando la extracción de características como atenuación espectral, Centroide espectral, Flujo espectral, MFFC, Cruce por cero generando un vector de características con media y desviación estándar de los parámetros.

Los diferentes sistemas multi experto para la clasificación automática de géneros de música (Paradzinets, Harb, y Chen, 2009), donde han propuesto varios sistemas. En su mayoría, estos sistemas son una adaptación de un clasificador de audio general a la tarea de clasificación de género musical. Utilizan un análisis solo de señal. planteando la hipótesis de que un enfoque de análisis de señal para la clasificación de género musical tiene sus limitaciones. En primer lugar, debido a la complejidad de la tarea y en segundo lugar debido a la inadecuación de las características de la señal generalmente utilizados. Por lo tanto, proponen el utilizar diferentes características de títulos de música: relacionados con el timbre, relacionados con el ritmo y relacionados con el artista. Se propone una arquitectura clasificadora multi-experta para ser utilizada con diferentes características, en el caso del experto acústico, la aplicación del modelado del filtro auditivo humano antes del Piecewise Gaussian Modeling aporta una ligera mejora de las tasas de clasificación en comparación con el PGM básico. Una mejora más significativa del rendimiento se logra utilizando Multi PGM-MLP. Sin embargo, estas mejoras no son estadísticamente significativas. El experto

rítmico utiliza el histograma de latido 2D combinado con una medida de similitud básica entre histogramas y da resultados satisfactorios. Sin embargo, el experto rítmico encuentra sus límites en la clasificación de géneros como Jazz, Metal, Rock (Pop Rock).

(López y Molina, 2017), plantean el crecimiento de los datos digitalizados de las diferentes ramas del saber, lo cual ha propiciado la creación de herramientas que realicen el análisis automático de estos contenidos. En ese contexto, los contenidos musicales no son ajenas al fenómeno, donde la necesidad de organización de la música, condujo a la creación de géneros musicales. Es donde la clasificación de géneros se considera un proceso subjetivo el cual se ve afectado por diversos factores; es por esto que no existe una única forma de clasificación de un género musical, teniendo en cuenta esta premisa plantean la clasificación de subgéneros de música electrónica mediante la extracción del vector de características planteado por (Tzanetakis y cols., 2002), donde presentan la hipótesis de que para el reconocimiento automático de un género se debe tener en cuenta que se presentan ciertas características compartidas entre estas, y de las cuales se pueden generar un modelo predictivo a partir del reconocimiento de patrones en las señales; además, plantean el uso de grafos de confusión como complemento gráfico en el desarrollo del algoritmo y visualización de los resultados de la clasificación de los 23 subgéneros de música electrónica con lo cual obtuvieron un 50 % de acierto en la clasificación de los subgéneros de este tipo de música.

6.2.2. Clasificación en Colombia

Del mismo modo, (García, Pacheco, y Mancera, 2005) plantea que el manejo de las bibliotecas digitales hoy en día ha sido un campo de estudio para el análisis y procesamiento de señales de audio debido al crecimiento en el intercambio en la red de computadores de grandes bases de datos, creando la necesidad de generar sistemas inteligentes que clasifiquen esta información, sin embargo, los estudios que se han realizado para determinar las características que constituyen un estilo musical busca determinar que propiedades son importantes y cuáles son las metodologías de aprendizaje con un porcentaje de error mínimo en el procesamiento de la información, en las cuales las muestras son tomadas de las canciones en formato .WAV con una duración de 1.49 segundos en canciones con un muestreo de 44.1 kHz, donde divide las características en tres grupos: El primero es el análisis en el tiempo haciendo uso de la correlación; El segundo grupo se basa en el análisis en frecuencia que utiliza la transformada de Fourier; El último grupo busca la extracción de dos aportes rítmicos independientes haciendo uso de la transformada de wavelet donde escogen la función sombrero mexicano como función para la extracción de los aportes rítmicos. Para el aprendizaje de la máquina plantean un perceptrón multinivel y redes de

funciones de base radial (RBF), donde el método de aprendizaje se encuentra supervisado por un algoritmo Adaboost el cual genera un alto nivel de certeza en la clasificación en los géneros escogidos (Salsa, merengue, vallenato)

(Muñoz, 2016) por otro lado, plantea que los géneros musicales tienen características comunes que comparten los miembros, características como la instrumentación, la textura tímbrica, contenido rítmico y contenido armónico, de esta manera expone la caracterización de cuatro géneros bambuco, cumbia, porro, vallenato; de un total de 85 canciones realizando una ilustración del análisis cualitativo y cuantitativo del proceso realizado para el vector de diecinueve características en la textura tímbrica, implementación del histograma de ritmo en cada una de las canciones y la extracción de la matriz de similitud o cuadro de ajedrez con el cual obtuvo las características principales de cada géneros para el grupo de canciones, de manera que con el estudio en la caracterización de la música folclórica colombiana, encontró varios factores, en donde establece un grado de incertidumbre en el entendimiento y universalidad del folclor colombiano permitiendo entender que la cumbia y el porro tiene tonos brillantes, los bambucos presentan tonos menos brillantes permitiendo encontrar una diferencia para la clasificación mediante la extracción de características.

De manera que (Pedraza, Vitola, Sepulveda, y Martinez, 2013), plantean un rápido algoritmo de recuperación de audio basado en contenido, donde, hacen uso de la toma de huella digitales en la identificación de audios de amplio espectro, teniendo presente que la recuperación de audios musicales tiene dos retos principales la robustez y la velocidad de respuesta del algoritmo, la extracción de características de frecuencia del audio y una función hash determinaron un bajo nivel de error y un tiempo de respuesta menor con respecto a otras técnicas, donde la aplicabilidad se pueden destinar a aplicaciones en tiempo real para el monitoreo de estaciones de radio; donde la recuperación de contenidos de audio por el reconocimiento de pistas de audio con derecho de autor en diferentes medios, en el que el uso de la huella digital permite la búsqueda más rápida y eficiente en bases de datos, planteando que aun con un alto grado de confiabilidad de un algoritmo con huella digital enfrenta problemas para la identificación de las pistas de audio al haber un cambio en la velocidad, variaciones de tono, ruido, más sin embargo, la carga computacional requerida es menor a otras técnicas; en el proceso experimental observaron la reducción del rendimiento del algoritmo al agregar ruido blanco a las pistas, a lo cual las pistas de 5 segundos fueron las de mayor afectación en la variación de la sensibilidad con respecto a las pistas de 10, 20 y 30 segundos donde se observó una robustez contra el ruido en la identificación con respecto a la base de datos, además en el proceso de análisis de respuesta del algoritmo se observó una mejora en el rendimiento con el trabajo en plataformas paralelas.

Ahora bien, el uso continuo de medios electrónico como teléfonos móviles o reproductores de música han permitido que la accidentalidad vehicular tenga cerca del 14 % de mortalidad en los peatones o ciclistas como lo explica (Giraldo-Guzmán, Marrugo, y Contreras-Ortiz, 2016), en donde, proponen mediante el análisis de señales acústicas generadas por los vehículos buscan calcular la velocidad de aproximación, planteando la adquisición de señal, pre- procesamiento, extracción de características y el ingreso a la red neural, la cual realiza el aprendizaje de las diferentes señales que ingresan, el proceso lo realizan con Matlab R2010, donde plantean el modelamiento del sistema con los ruidos generados por el motor, neumáticos, sistema de escape entre otras piezas del carro, determinando cuando un vehículo se encuentra en movimiento o se encuentra en velocidad de enciendo mediante la codificación de predicción lineal y el uso del coeficiente de Mel MFCC para la envolvente del espectro del vehículo, donde obtuvieron que el ZCR disminuye a altas velocidades en relación señal / ruido; teniendo presente esto estimaron que las redes neuronales pueden modelar diferentes tipos de fenómenos cuando se diseñan y entrenan adecuadamente.

6.2.3. Cultura digital

Ahora bien, la sociedad se encuentra en una multidimensionalidad sin fronteras, en las cuales la juventud se identifica con el avance tecnológico como lo manifiesta (Rodríguez-Quiles y cols., 2011), sin embargo, plantea que las culturas juveniles se pluralizan y diferencian en como entran en contacto con los medios tecnológicos, donde las comercialización es flexible, como en el caso del fenómeno musical, con todo esto el interés individual o grupal permite a la mayoría de jóvenes tener una cercanía con estos medios de expresión generando una educación musical a la cual se denomina generación digital, así las cosas, se pueden denominar a este segmento de población *homo ludens digitalis* debido a la interacción que realizan los jóvenes con la cultura musical que se encuentra de moda. Teniendo presente que el proceso es paulatino e imparable donde la tecnologización y comercialización de medios culturales es cada vez menor, con relación a la transmisión de información adulto-niño o adulto-adolescente donde el desarrollo personal está sujeto a las competencias que brindan las nuevas tecnologías donde la conexión web se convierte en una plataforma asociada a un inmenso negocio internacional en una sociedad consumista en el cual los jóvenes no escatiman en los gastos a la hora de encontrarse a la moda en la cultura estética que impone cultura digital, siendo la adolescencia un nicho cultural el cual busca personas e ideas en las cuales afianzar su crecimiento personal siendo la música y la moda en espacios digitales un símbolo de estatus como el comparar sus relaciones interpersonales con el número de amigos en las redes sociales; como se ha visto la aceleración global en la que nos encontramos ha formado una cultura digital e identidad personal de acuerdo con el

contacto con la internet, de esta manera, las sub-identidades se presentan como grupos a los cuales se le es difícil una comprensión de cómo ven el mundo e interactúan con él, siendo la interconectividad un medio en el cual se pueden generar ofertas de participación ciudadana en las cuales se vinculan a los jóvenes mediante juegos o eventos musicales online.

Dentro de este contexto, (Mejía Gallego, 2012) plantea el diseño de un material interactivo, con contenido didáctico desarrollado en cinco ritmos folclóricos; debido a la falta de motivación de los niños hacia la música tradicional colombiana en el cual, busco desarrollar las habilidades visuales y auditivas a través de juegos los cuales logran estimular una cercanía de los niños a la música tradicional, donde la percepción visual auditiva toman factor notable en el acercamiento de los infantes a la música tradicional teniendo en cuenta que la percepción visual es un fenómeno con el cual se convive a diario de manera que las sensaciones que un objeto genera se convierte en algo común es donde busca la manera de generar actividades que logren la estimulación de los niños este acompañado por la percepción auditiva que genera la música tradicional la cual acompaña el esquema gráfico de los juegos y actividades interactivas con las cuales busco fortalecer la cultura tradicional en el grupo de niños. Con la definición de los ritmos tradicionales enmarcados en cada una de las regiones cartográficas mediante el plan del ministerio de cultura en cual le permitió conocer la actualidad y practicas musicales enmarcadas dentro de un marco geográfico dando el formato instrumental, de manera que le permitió mediante las actividades lúdico recreativas enseñarles a los niños sobre la ubicación e identidad de la música de cada región.

(Melo Ángel, s.f.) plantea a Colombia como un país pluriétnico, donde las características que se expresan en los diversos géneros del territorio nacional, contiene la historia de las regiones y su población de manera que hacen parte de la cotidianidad a lo largo del territorio, por otro lado, el patrimonio musical genera diferentes sentimientos de identidad en la población regional, de manera que la salvaguardia del patrimonio musical es un tema de interés de la academia en sus diferentes ramas del saber. De acuerdo con la UNESCO, el ámbito patrimonial ha tomado un renovado protagonismo en la investigación, identificación y preservación como estrategia principal, donde la creación de escuelas ayuda a orientar el proceso mediante la cartografía musical a la enseñanza y preservación de los ritmos tradicionales de manera apropiada en donde la documentación musical no se encuentre desarticulada de la misma buscando salvaguardar el patrimonio musical; con el reconocimiento multicultural en la constitución de 1991 en la cual se busca la promoción y protección de la diversidad cultural de manera que la difusión de la documentación musical para que la población externa conozcan y entienda el significado y valor de quien

produjeron estos medios. Si bien presenta una mirada acrítica al patrimonio musical, la fosilización de la música por el desarrollo de las categorías musicales académicas en las cuales se hace uso de la música como instrumento político de manera que surgen problemas al implementar la salvaguarda de documentación musical; el reconocer la multiculturalidad de Colombia genera un alto grado de incertidumbre de como instituciones como el CDM, ha buscado salvaguardar la documentación musical teniendo presente el marco jurídico con el cual se preserva y tiene acceso a la documentación musical, los derechos de autor, la recuperación de archivos sonoros y catalogación del documento.

Evidentemente (Quevedo Urrea, 2011), coopera en el análisis y recuperación de la documentación musical de Colombia, de manera que los técnicos en bibliotecología han encontrado de manera general, el recuperar estos medios como algo fuera de su rama del saber por lo que se genera una exclusión y pérdida del conocimiento del patrimonio nacional, donde la documentación, conformada por manuscritos, audios, la mayoría de los medios audiovisuales en formato análogo presentan dificultad en la manera en que se clasifican, de manera que al ampliar la formación y practica en documentación musical permite la evolución en la interdisciplinaridad en el usos de la documentación musical, dando experiencias a una población multicultural a la cual el desarrollo y perspectivas le han generado una pérdida en el patrimonio nacional, con el proyecto de cartografía musical la cual es una iniciativa para contrarrestar esta pérdida busca una formalidad en la circulación y estrategias de investigación apoyado en el desarrollo de las TIC.

7. DISEÑO METODOLÓGICO

En este capítulo, se da una explicación descriptiva del proceso e implementación en el lenguaje octave para la extracción de características, además, de la implementación del algoritmo en R para aprendizaje y predicción de los géneros. Para la consecución de los objetivos mencionados anteriormente, es necesario el planteamiento de un modelo predictivo para la clasificación automática de géneros de música folclórica colombiana. El trabajo debe cumplir las siguientes fases:

- Estandarizar los cuatro géneros folclóricos colombianos a nivel del vector de características.
- Identificar el tipo de aprendizaje adecuado para la clasificación de los cuatro géneros folclóricos colombianos a partir de las características.
- Establecer una técnica para la clasificación de los cuatro géneros folclóricos colombianos a partir de sus características.
- Evaluar el porcentaje de precisión del algoritmo en la clasificación automática de los géneros folclóricos colombianos

7.1. Estandarización de los cuatro géneros folclóricos colombianos

Inicialmente se determinan las características que permiten una estandarización de cuatro géneros folclóricos, para ello se realizaron las siguientes tareas:

- Buscar un entendimiento inicial de los géneros folclóricos colombianos mediante las características que permitan la recolección de la información a analizar.
- Recolectar la mayor cantidad de piezas musicales para la biblioteca con la cual se va a realizar el análisis de las características.
- Crear un banco de datos de características con las muestras adquiridas que permitan el desarrollo del software.

7.1.1. Aproximación inicial

Como se expuso en la sección 6.1.9; y teniendo presente que se encuentran enmarcados en el plan nacional de desarrollo (Uribe, 2002) mediante el PNMC donde se proponen los 11

“ejes musicales” definidos a partir de las practicas más reconocías de cada región se obtiene que:

Bambuco Originario de la Región Andina cuyo compás es medido a 3/4 ó 6/8 comúnmente, donde la instrumentación está representada por Quiribillo, Esterilla, Chucho, Tambora, Flautas de caña, Riolina o dulzaina, Guitarra, Tiple, Requinto, Bandola andina. En sus inicios el bambuco se interpretó con bandola, tiple y guitarra. Su canto es preferentemente para ser interpretado a dueto de voces, de los cuales han dejado profunda huella por su gran calidad instrumental y vocal.

Currulao Originario de la Región Pacífica cuyo compás es medido a 6/8 comúnmente, donde la instrumentación está representada por la marimba además, tambora, Cununo hembra, Cununo macho. Su canto en el coro generalmente está a cargo de las mujeres y desarrolla versos reiterados en estribillo y fonemas sujetándose al proceso rítmico y dejando que la melodía del canto se dibuje sin relieve vocal. En la ejecución del currulao es posible observar las características propias de un rito sacramentalmente impregnado de fuerza ancestral y de contenido mágico

Joropo Originario de la Región de la Orinoquia cuenta con un gran carácter folclórico. Posee movimiento rápido a ritmo ternario mezclado con un vals; tienen por métrica una combinación de 6/8 y 3/4, aun cuando hay casos particulares donde figuran elementos de hasta 3/2. Como instrumento principal el Arpa acompañada por Bandola llanera, Cuatro, Maracas o capachos, Bandolín y Cirrampla. Un coplero es el cantador de coplas llaneras para el joropo, y generalmente es un hombre que se inspira en sus vivencias amorosas o de faenas en las sabanas para cantar o componer sus coplas.

Vallenato Originario de la Región Atlántica, la interpretación del ritmo esta determinada por el subgénero el cual puede ser de 4/4, 2/4, 6/8;, donde la instrumentación esta representada por el uso del acordeón como instrumento principal, acompañado por la caja, la guacharaca, la guitarra y el bajo eléctrico. Su canto se encuentra enmarcado en la interpretación del aire vallenato al que corresponde el cantautor, donde, el merengue es predominantemente africano; el paseo es más español y la puya es notoriamente indígena; no obstante en el curso de su constitución todos recibieron el influjo de la diversidad.

Teniendo esto presente y de acuerdo a (Educational, 2015) y a (Uribe, 2002), donde se desarrolla el genero vallenato como patrimonio que aporta Colombia a la humanidad se

tiene presente para la clasificación como genero a los tres aires vallenatos principales como lo son el paso, la puya y el merengue los cuales se clasifican como un genero general que es el vallenato.d

7.1.2. Audioteca

En la búsqueda de audioteclas que permitieran adquirir las piezas musicales para el análisis, se encuentra que en medio digital el único genero el cual se encuentra por etiqueta es el vallenato en plataformas como Spotify (Spotify, 2018) donde, se hace hincapié en que los muestras musicales deben incluir otros datos, no solo la canción en sí. Quiere que los artistas y los estudios discograficos anoten cosas como el género, el estado de ánimo y otros datos, incluidos elementos como los instrumentos utilizados, ya sea una portada, la cultura a la que pertenece la canción y más. Sin embargo, es una plataforma en la cual géneros folclóricos no tienen etiquetas de búsqueda. Teniendo esto claro se busca ayuda en la plataforma Youtube para la adquisición de los géneros los cuales son clasificados mediante etiquetas de genero teniendo presente que estas etiquetas tiene un grado de incertidumbre por el desconocimiento de quién clasifica las piezas musicales, se busca la colaboración de un experto en musica para que verifique las etiquetas. Para la descarga de las piezas musicales se hace uso del paquete “youtube-dl”(ANEXO A) nativo de Python mediante el cual se descargan las muestras son archivos en formato “.wav”, con frecuencia de muestreo 44100 estéreo, ya que es el estándar utilizado a la hora de realizar extracción de características; el grupo para clasificación es de 88 por genero para un total de 352 canciones de las cuales 64 hacen parte del grupo de enseñanza para la clasificación.

7.1.3. Banco de datos

Teniendo presente que las señales de sonido se pueden clasificar de manera muy amplia en sonidos ambientales, sonidos artificiales, habla y música. Una gran clase de sonidos interesantes varía en el tiempo en su naturaleza con información codificada en forma de secuencias temporales de eventos de sonido atómico. Por ejemplo, el habla se puede ver como una secuencia de teléfonos y la música como un patrón de notas en evolución. Un evento de sonido atómico, o un solo gestalt, puede ser una señal acústica compleja descrita por un conjunto específico de propiedades temporales y espectrales. Los ejemplos de eventos de sonido atómico incluyen sonidos cortos, como un portazo, y sonidos de textura más larga y uniforme, como el constante golpeteo de la lluvia. Las propiedades temporales de un evento de audio se refieren a la duración del sonido y cualquier modulación de amplitud, incluyendo el aumento y la caída de la envolvente de amplitud de la forma de onda. Las

propiedades espectrales del sonido se relacionan con sus componentes de frecuencia y sus fuerzas relativas.

El proceso de adquisición del banco de datos es tomado de las muestras de musica descargada, se realiza mediante el proceso desarrollado por (Muñoz, 2016) en donde se implementa una caracterización de cuatro géneros folclóricos colombianos mediante el vector de características presentado por (Tzanetakis y cols., 2002) donde se tienen características a nivel tímbrico, armónico y tonal y en donde (Muñoz, 2016) permite extraer las características con ayuda de algoritmos en lenguaje Octave, estos códigos presentaban vector de características del segundo de análisis en relación a media y desviación estándar en las ventanas de análisis para resumir y obtener las características resultantes de la pieza musical, de manera que para el desarrollo del proyecto se ve la necesidad de obtener los datos de manera directa y se modifican los códigos para la obtención de 98 observaciones por pieza musical para la clasificación que se realizara en el proceso de aprendizaje de máquina.

Teniendo presente esto las características físicas son parámetros de señal de bajo nivel que capturan aspectos particulares de las propiedades temporales o espectrales de la señal. Aunque algunas de las características están motivadas por la percepción, las clasificamos como características físicas, ya que se calculan directamente a partir de las amplitudes de la forma de onda del audio o los valores espectrales de corto tiempo correspondientes. Las características físicas ampliamente aplicadas se discuten a continuación. En las siguientes ecuaciones, el subíndice r indica el marco actual, de modo que $x_r[n]$ son las muestras del segmento de datos de longitud N correspondiente al marco actual.

Los algoritmos modificados buscando optimizar el proceso en la extracción y organización de los datos de ingreso al algoritmo de aprendizaje se hace uso del algoritmo `cell2csv` y `cell2file` como se muestran a continuación:

Centroide Espectral: Dado por la ecuación 31.

$$C_t = \frac{\sum_{n=1} M_t[n] * n}{\sum_{n=1} M_t[n]} \quad (31)$$

Donde $M_t[n]$ es la magnitud de la transformada de Fourier en el marco t y la frecuencia en el comportamiento n , para La modificación del código es:

```
1 % Copyright (C) Ingenieria Electronica. 2018 UCundinamarca
2 %
3 % Calcula el centroide espectral de @var{vector}
```

```

4 %
5 % Author: Sebastian Munoz <sebastianmg.info@gmail.com>
6 % Created: 2016
7
8 % modified by: Jose Zamora <learningtesis@hotmail.com>
9 % Date: 2018
10 function [Ct]=centroide(Tw,Ts,fs,vector)
11
12 %% verificar parametros
13 Nw=round(1E-3*Tw*fs);      % duracion del marco en muestras
14 Ns=round(1E-3*Ts*fs);      % duracion del cambio de marco en muestras
15 nfft=2^nextpow2(Nw);       % Longitud del analisis FFT
16
17 % Enmarcado y ventanamiento (marcos como columnas)
18 frames=vec2frames(vector,Nw,Ns);
19 % Calculo del espectro de magnitud (como vectores columna)
20 MAG=abs(fft(frames,nfft,1));
21
22 m=size(MAG);
23 t=1:m(2);
24 n=1:m(1);
25 n=n';
26 mag_n=(MAG).*(n);
27
28 num=sum(mag_n);
29 den=sum(MAG);
30 Ct=num./den; % Centroide Espectral
31
32 end

```

Listing 1: Centroide

Atenuación Espectral: Para la atenuación se busca un porcentaje cerca del 85 % mediante la ecuación 32:

$$\sum_{n=1}^{R_t} M_t[n] = 0,85 * \sum_{n=1}^N M_t[n] \quad (32)$$

El algoritmo modificado es:

```

1 % Copyright (C) Ingenieria Electronica. 2018 UCundinamarca
2 %
3 % Calcula la Atenuacion espectral de @var{vector}

```

```

4 %
5 % Author: Sebastian Munoz <sebastianmg.info@gmail.com>
6 % Created: 2016
7
8 % modified by: Jose Zamora <learningtesis@hotmail.com>
9 % Date: 2018
10
11 function[Ae]=atenuacion(Tw,Ts,fs,vector)
12
13 Nw=round(1E-3*Tw*fs);      % duracion del marco en muestras
14 Ns=round(1E-3*Ts*fs);      % duracion del cambio de marco en muestras
15
16 nfft=2^nextpow2(Nw);      % Longitud del analisis FFT
17
18 % Enmarcado y ventanamiento (marcos como columnas)
19 frames=vec2frames(vector,Nw,Ns);
20
21 % Calculo del espectro de magnitud (como vectores columna)
22 MAG=abs(fft(frames,nfft,1));
23
24 Ae=0.85.*(sum(MAG));      % Atenuacion Espectral
25
26 end

```

Listing 2: Atenuacion espectral

Flujo espectral: Se busca normalizar la magnitud del marco actual t , y el anterior $t-1$ mediante el uso de la transformada de Fourier. Mediante la ecuación 33:

$$F_t = \sum_{n=1}^N (N_t[n] - N_{t-1}[n])^2 \quad (33)$$

El algoritmo modificado:

```

1 % Copyright (C) Ingenieria Electronica. 2018 Ucundinamarca
2 % Calcula el flujo espectral de @var{vector}.
3 % Author: Sebastian Munoz <sebastianmg.info@gmail.com>
4 % Created: 2016
5 % modified by: Jose Zamora <learningtesis@hotmail.com>
6 % Date: 2018
7 function [Ft_sum] = flujo(Tw, Ts, fs, vector)
8
9 Nw = round( 1E-3*Tw*fs );      % duracion del marco en muestras

```

```

10 Ns = round( 1E-3*Ts*fs );      % duracion del cambio de marco en muestras
11 nfft = 2^nextpow2( Nw );      % Longitud del analisis FFT
12
13 % Enmarcado y ventanamiento (marcos como columnas)
14 frames = vec2frames(vector, Nw, Ns);
15 % Calculo del espectro de magnitud (como vectores columna)
16 MAG = abs(fft(frames,nfft,1));
17
18 m=size(MAG);
19 t=1:m(2);
20 n=1:m(1);
21 n=n';
22
23 Ft=zeros(m);
24 mayor=max(MAG);
25
26 for i=2:m(2)
27     Nt_1=(MAG(:,i))/mayor(i);      % se normaliza con el mismo numero???
28     Nt_2=(MAG(:,i-1))/mayor(i-1);
29     Ft(:,i)=(Nt_1-Nt_2).^2;      % Flujo Espectral
30 end
31
32 Ft_sum=sum(Ft);
33
34 end

```

Listing 3: Flujo espectral

Cruce por cero: Se implementa la detección de cruce por cero tomada por (Muñoz, 2016), en el cual se usa la ecuación 34:

$$Z_t = \frac{1}{2T_W} \sum_{n=1}^{T_w} |sgn(x_t[n]) - sgn(x_t[n-1])| \quad (34)$$

El algoritmo modificado:

```

1 % ## Copyright (C) Ingenieria Electronica. 2018 Ucundinamarca
2
3 % Deteccion del cruce por cero
4
5 % ## Author: Sebastian Munoz <sebasmg.reggae@gmail.com>
6 % ## Created: 2016
7 % ## Modified by: Jose Zamora <learningtesis@hotmail.com>

```



```

8 % ## Date: 2018
9 function [Zt_sum] = cruce(Tw, Ts, fs, vector)
10
11 Nw = round( 1E-3*Tw*fs );    % frame duration (samples)
12 Ns = round( 1E-3*Ts*fs );    % frame shift (samples)
13
14 nfft = 2^nextpow2( Nw );      % length of FFT analysis
15
16 % Framing and windowing (frames as columns)
17 frames = vec2frames(vector, Nw, Ns);
18
19 % Magnitude spectrum computation (as column vectors)
20 %MAG = abs(fft(frames,nfft,1));
21
22 m=size(frames);
23 t=1:m(2);
24 n=1:m(1);
25 n=n';
26
27 Zt=zeros(1,m(2));
28
29 for i=2:m(1)
30     m1=frames(i,:);
31     m2=frames(i-1,:);
32     s1=signo(m1);
33     s2=signo(m2);
34     Zt(i,:)=abs(s1-s2);
35 end
36
37 Zt_sum=1/2*(sum(Zt));
38
39 end
40
41 function s = signo(x)
42 s=x>0;
43 end

```

Listing 4: Cruce por cero

Vector textura tímbrica: Este algoritmo se usa para extraer las características por ventanas de análisis en un vector donde se almacenan las 9 características de análisis como se muestra a continuación:

```

1
2 function vec_caract = vector_tex_tim(nombre,t=[x1 x2],Tw,Ts)

```

```

3 if !ischar(nombre)
4     error ('no es un nombre de archivo');
5 end
6
7
8 alpha = 0.97;           % coeficiente de pre-enfasis
9 M = 20;                 % Cantidad de canales de filtros
10 C = 12;                % Numero de coeficientes cepstrales
11 L = 22;                % Parametro cepstral sine lifter
12 LF = 300;              % Limite de frecuencia inferior (Hz)
13 HF = 3700;             % Limite de frecuencia superior (Hz)
14
15     [vector,fs,nbits]=wavread(nombre,100); % lectura del audio para
saber la frecuencia fs
16     [vector,fs,nbits]=wavread(nombre,t*fs);
17     vector=vector(:,1);
18
19     [Ct]=centroide(Tw,Ts,fs,vector);
20     [Ae]=atenuacion(Tw,Ts,fs,vector);
21     [Ft]=flujo(Tw,Ts,fs,vector);
22     [Zt]=cruce(Tw,Ts,fs,vector);
23     [MFCCs,FBES,frames]=mfcc(vector,fs,Tw,Ts,alpha,@hamming,[LF HF],M,C
+1,L);
24     MFCCs5=MFCCs(1:5,:);
25     L_E = low_e(Tw, Ts, fs, vector);
26     vec_caract =[Ct;Ae;Ft;Zt;MFCCs5];
27
28 end

```

Listing 5: Vector textura tímbrica

Función textura tímbrica: Mediante esta función se llama el algoritmo vector textura tímbrica, adicional se organizan los datos extraídos con nombre de la canción y se realiza llamado de la biblioteca mediante la función `glob` como se observa a continuación:

```

1 Tw = 25;                % Duracion de la trama de analisis (ms)
2 Ts = 10;                % Cambio de trama de analisis (ms)
3 t1= 90;
4 t2= 91;
5
6 for i=1:4
7     if (i==1)
8         wav_file =glob('/home/jestebanza/Aprendizaje/audicion computacional/
canciones/bambuco/*.wav');
9         file1 =glob('/home/jestebanza/Aprendizaje/audicion computacional/
canciones/bambuco/*.wav');

```

```

10     w=length(wav_file);
11     vec_caract_bam=zeros(10,w*98);
12 elseif (i==2)
13     wav_file =glob('/home/jestebanza/Aprendizaje/audicion computacional/
14     canciones/currulao/*.wav');
15     file2 =glob('/home/jestebanza/Aprendizaje/audicion computacional/
16     canciones/currulao/*.wav');
17     w=length(wav_file);
18     vec_caract_cur=zeros(10,w*98);
19 elseif(i==3)
20     wav_file =glob('/home/jestebanza/Aprendizaje/audicion computacional/
21     canciones/joropo/*.wav');
22     file3 =glob('/home/jestebanza/Aprendizaje/audicion computacional/
23     canciones/joropo/*.wav');
24     w=length(wav_file);
25     vec_caract_jor=zeros(10,w*98);
26 elseif (i==4)
27     wav_file =glob('/home/jestebanza/Aprendizaje/audicion computacional/
28     canciones/vallenato/*.wav');
29     file4 =glob('/home/jestebanza/Aprendizaje/audicion computacional/
30     canciones/vallenato/*.wav');
31     w=length(wav_file);
32     vec_caract_va=zeros(10,w*98);
33 end
34
35     for canciones=1:w
36         wav=wav_file(canciones,1);
37         wav=char(wav);
38         vec_caract = vector_tex_tim(wav,t=[t1 t2],Tw,Ts);
39
40         if (i==1)
41             vec_caract_bam =[vec_caract];
42             vec_caract_bam=vec_caract_bam';
43             long=size(vec_caract_bam)(1);
44             for canc= 1:long
45                 printf("%0.10f,",vec_caract_bam(canc,:));
46                 printf("%s",char(wav_file(canciones,1)));
47                 printf("%s", ", bambuco");
48                 printf("\n");
49             end
50
51             elseif (i==2)
52                 vec_caract_cur=[vec_caract];
53                 vec_caract_cur=vec_caract_cur';
54                 long=size(vec_caract_cur)(1);
55                 for canc= 1:long
56                     printf("%0.10f,",vec_caract_cur(canc,:));

```

```

50     printf("%s",char(wav_file(canciones,1)));
51     printf("%s", ",currulao");
52     printf("\n");
53 end
54     elseif (i==3)
55         vec_caract_jor=[vec_caract];
56         vec_caract_jor=vec_caract_jor';
57         long=size(vec_caract_jor)(1);
58     for canc= 1:long
59         printf("%0.10f,",vec_caract_jor(canc,:));
60         printf("%s",char(wav_file(canciones,1)));
61         printf("%s", ",joropo");
62         printf("\n");
63     end
64     elseif (i==4)
65         vec_caract_va=[vec_caract];
66         vec_caract_va=vec_caract_va';
67         long=size(vec_caract_va)(1);
68     for canc= 1:long
69         printf("%0.10f,",vec_caract_va(canc,:));
70         printf("%s",char(wav_file(canciones,1)));
71         printf("%s", ",vallenato");
72         printf("\n");
73     end
74     end
75
76     end
77 end
78
79
80 csvwrite('textura_timbrica_bam.csv',[vec_caract_bam]);
81 csvwrite('textura_timbrica_cur.csv',[vec_caract_cur]);
82 csvwrite('textura_timbrica_jor.csv',[vec_caract_jor]);
83 csvwrite('textura_timbrica_va.csv',[vec_caract_va]);
84 csvwrite('textura_timbrica.csv',[vec_caract_bam;vec_caract_cur;
    vec_caract_jor;vec_caract_va]);
85 cell2csv('nombres.csv',[file1;file2;file3;file4]);

```

Listing 6: Textura timbrica

Algunos algoritmos usados en el proyecto se encuentran en el ANEXO B, con los cuales se extraen los nombres y se realiza la conversión de características de los datos.

Donde las formas de onda de audio pueden ser periódicas o aperiódicas. A excepción de la senoide simple, las formas de onda de audio periódicas son tonos complejos que comprenden una frecuencia fundamental y una serie de armónicos o múltiplos de la frecuencia

fundamental. Las amplitudes relativas y las fases de los componentes de frecuencia influyen en el sonido color.^o timbre. Las formas de onda de aperiódica, por otro lado, pueden estar compuestas de tonos sinusoidales relacionados o ruido en forma de frecuencia. En general, un sonido puede exhibir propiedades espectrales tanto de tono como de ruido y éstas influyen en la calidad percibida. El habla se caracteriza por la alternancia de regiones tonales y ruidosas con duraciones de tono correspondientes a segmentos vocales que ocurren a una velocidad silábica más regular. Por otro lado, la música, al ser una secuencia melódica de notas, es muy tonal en su mayor parte, con una frecuencia y una duración fundamentales que varían en un amplio rango.

De manera que las señales de sonido son básicamente estímulos físicos que son procesados por el sistema auditivo para evocar sensaciones psicológicas en el cerebro. Es apropiado que las propiedades acústicas sobresalientes de un sonido sean las que son importantes para la percepción humana y el reconocimiento del sonido. La percepción auditiva se ha estudiado desde 1870, la época de Helmholtz. Los sonidos se describen en términos de los atributos perceptivos de tono, sonoridad, duración subjetiva y timbre. Se sabe que el sistema auditivo humano realiza el análisis de frecuencia de los sonidos para alimentar las funciones cognitivas de nivel superior. Cada una de las sensaciones subjetivas se correlaciona con más de una propiedad espectral (por ejemplo, contenido tonal) o propiedad temporal (por ejemplo, el ataque de una nota tocada en un instrumento) del sonido. Dado que tanto las propiedades espectrales como las temporales son relevantes para la percepción y el conocimiento del sonido, solo es apropiado considerar la representación de las señales de audio en términos de una descripción conjunta en el tiempo y la frecuencia.

Donde las aplicaciones tecnológicas importantes del procesamiento de señales de audio digital son la compresión de datos de audio, la síntesis de los efectos de audio y la clasificación de audio hacen uso de la extracción de características temporales. Si bien la compresión de audio ha sido la aplicación más prominente del procesamiento de audio digital en el pasado reciente, la creciente importancia de la gestión de contenido multimedia es ver las crecientes aplicaciones de procesamiento de señales en la segmentación y clasificación de audio. La clasificación de audio es parte del problema mayor del manejo de datos audiovisuales con aplicaciones importantes en bibliotecas digitales, producción profesional de medios, educación, entretenimiento y vigilancia. El reconocimiento del habla y del hablante pueden considerarse problemas clásicos en la recuperación de audio y han recibido décadas de atención en la investigación. Por otro lado, los archivos en rápido crecimiento de música digital en Internet ahora están llamando la atención sobre problemas más amplios de navegación y recuperación no lineal que utilizan formas más naturales de interactuar con datos multimedia, incluida, principalmente, la música. Dado que los registros de audio (a

diferencia de las imágenes) se pueden escuchar solo de forma secuencial, una buena indexación es valiosa para una recuperación efectiva. Escuchar clips de audio puede ayudar a navegar el material audiovisual más fácilmente que la visualización de escenas de video. La clasificación de audio también es útil como una interfaz para sistemas de compresión de audio donde la eficiencia de la codificación y la transmisión se facilita al hacer coincidir el método de compresión con el tipo de audio, como por ejemplo, el habla o la música.

7.2. Identificación los métodos de aprendizaje

Posteriormente se procedió a identificar los métodos de aprendizaje que permitieron un alto grado de precisión en la clasificación de los cuatro géneros, para esto se llevarán a cabo las siguientes actividades:

- Realizar una comparación entre los métodos de aprendizaje de máquina buscando un alto grado de desempeño.
- Mediante el análisis de la comparación generar una tabla comparativa de cuál es el mejor método de aprendizaje que permita un alto grado de precisión en la clasificación

7.2.1. Tabla comparativa métodos de aprendizaje

En el proceso de elección del método de aprendizaje se comprende que tipo aprendizaje se ajusta a la clasificación deseada teniendo en cuenta ciertos parámetros y características extraídas para el análisis y clasificación de manera que la Tabla 6 da un análisis rápido de los diferentes métodos.

Algoritmo Aprendizaje	Velocidad de predicción	Velocidad de entrenamiento	Uso de memoria	Ajuste requerido	Evaluación general
Regresión logística lineal	Rápido	Rápido	Pequeño	Mínimo	Bueno para pequeños problemas con límites de decisión lineales
SVM lineal	Rápido	Rápido	Pequeño	Mínimo	Bueno para pequeños problemas con límites de decisión lineales
Arboles de decisiones	Rápido	Rápido	Pequeño	Algunos	Buena generalista, pero propenso a sobreajuste
Regresión logística no lineal	Lento	Lento	Mediano	Algunos	Bueno para muchos problemas binarios y maneja bien datos de alta dimensión
SVM no lineal	Lento	Lento	Mediano	Algunos	Bueno para muchos problemas binarios y maneja bien datos de alta dimensión
Vecino más cercano	Moderado	Mínimo	Mediano	Mínimo	Baja precisión, pero fácil de usar e interpretar
Naïve Bayes	Rápido	Rápido	Mediano	Algunos	Ampliamente utilizado para texto incluyendo filtrado de spam
Conjuntos	Moderado	Lento	Varía	Algunos	Alta precisión y buen rendimiento para conjuntos de datos de tamaño pequeño a mediano
Red neuronal	Moderado	Lento	Medio a grande	Muchos	Popular para clasificación, compresión, reconocimiento y previsión

Tabla 6: Tabla método de aprendizaje

Con respecto a los algoritmo de clasificación se tiene la distribución de parámetros para la selección del algoritmo que permitió realizar la organización de las muestras musicales, teniendo en cuenta, la velocidad de predicción, velocidad de entrenamiento, uso de recursos de memoria, adicionalmente del ajuste requerido y las aplicaciones comúnmente usadas con estos algoritmos obteniendo una evaluación general de los mismos, donde a primera vista se descartan los algoritmos de regresión logística tanto lineal como no lineal, debido a que trabajan con grupo de datos pequeños además que presenta una velocidad de predicción lenta y un ajuste del modelo lento y extenso. Por otro lado, algoritmos como los árboles de decisiones requieren la detección de patrones con anterioridad como punto de comparación lo cual lo hace lento. Por último el algoritmo de redes neuronales junto con el algoritmo de conjuntos, son algoritmos con un costo de aprendizaje alto, un ajuste complicado debido a las numerosas capas que se deben desarrollar para obtener un grado de acierto alto. De manera que, algoritmos de agrupación o percepción del grupo como k -NN, clasificador bayesiano ingenuo y los clasificadores vector soporte de máquina son posibles algoritmos para la selección y clasificación de las muestras musicales como se observara en la sección de resultados.

7.2.2. Comparación métodos de aprendizaje

Como se observa en la figura 2 el ingreso de los diferentes datos permite entender que tipo de algoritmo de aprendizaje es el correcto para realizar la clasificación o predicción de una clase o grupo. De manera que para la elección de un método de aprendizaje se deben tener en cuenta ciertos parámetros como son:

- Número de observaciones de entrenamiento.
- Dimensionalidad del espacio de atributos o características.
- Los grupos son linealmente separables.
- Las características son independientes.
- Las características dependa linealmente del grupo de clasificación.
- El no tener un sobreajuste en el modelo.
- Velocidad de clasificación.
- Rendimiento en la clasificación.

Teniendo presente estos parámetros se aplica el principio la navaja de Ockham (Wikipedia, 2018a) donde se utiliza el algoritmo menos complicado como parte inicial y donde puede hacer frente a las necesidades; por otro lado se puede llegar a un algoritmo más complicado si es estrictamente necesario debido a la no linealidad en los parámetros de entrada.

k-vecino cercanos Es uno de los algoritmos más simples, sin embargo, es un clasificador robusto y versátil, el cual normalmente es el punto de referencia de algoritmos complejos como lo son redes neuronales y vectores de soporte (SVM). No obstante su simplicidad, k-NN puede superar a los clasificadores más potentes. Este algoritmo se basa en la variación del valor k al momento de analizar los k datos más cercanos (González, 2018).

Las ventajas de este algoritmo es el no ser paramétrico debido a que no realiza suposiciones explícitas sobre la forma funcional de los datos; es un algoritmo simple para explicar, comprender e interpretar; posee una alta precisión.

Las desventajas de este algoritmo es el ser insensible a los valores atípicos por consiguiente la precisión puede verse afectada por el ruido o las características irrelevantes; es un algoritmo que se basa en instancias por tanto no aprende un modelo, en su lugar escoge memorizar las instancias de entrenamiento que se utilizaran posteriormente en la fase de predicción; su costo computacional es alto debido a que almacena todos los datos de entrenamiento.

Clasificador Bayesiano ingenuo: Es un clasificador probabilístico que realiza la clasificación haciendo uso de la regla de decisión Máximo A Posteriori en un entorno bayesiano. También se puede representar utilizando una red bayesiana muy simple (Brownlee, 2016).

Las ventajas de este algoritmo está en el desacople de las clases de distribución condicional, permitiendo estimar de forma independiente; es un algoritmo de clasificación para problemas de clasificación binarios (de dos clases) y de clases múltiples. La técnica es más fácil de entender cuando se describe con valores de entrada binarios o categóricos.

Las desventajas del algoritmo es la falta de capacidad para comprender los parámetros fuera de las distribución probabilística; adicional la suposición de independencia de los atributos generando un fallo en la estimación de una clase.

Vectores de soporte Es un algoritmo que busca la correlación de los datos en un espacio de características grandes dimensiones de forma que las observaciones se puedan categorizar incluso si los datos no se puede separar linealmente (Center, 2018).

Las ventajas del algoritmo, es un clasificador en base a una sólida teoría. Donde, con los teoremas de minimización de riesgo se busca realizar un estado del arte del aprendizaje estadístico; de manera que se pueden aplicar a datos representados en cualquier espacio de Hilbert.

Las desventajas del algoritmo se tiende a volver complejo el kernel a utilizar con un grupo de datos demasiado grande; ineficientes para entrenar debido a la complejidad en los datos y al kernel seleccionado.

7.3. Técnica para la clasificación

Una vez identificados los métodos de aprendizaje de máquina se realizan pruebas para garantizar un funcionamiento óptimo, de esta manera se contemplan las siguientes pruebas para establecer el método de aprendizaje.

- Inspección en la clasificación de los cuatro géneros mediante los métodos de aprendizaje para la definición de la técnica.
- Determinar cual método de aprendizaje tiene la mejor velocidad de respuesta de clasificación sujeta a la capacidad de procesamiento del equipo.
- Verificación del proceso de clasificación mediante el análisis de las características obtenidas en el método de aprendizaje para el diseño del software de audición computacional.

7.3.1. Inspección en la clasificación para la definición de la técnica

Uno de los aspectos más importante al momento de realizar un algoritmo de aprendizaje de máquina es el tener en cuenta el tamaño, la calidad y la naturaleza de los datos adicional tener clara la respuesta que se busca obtener del algoritmo, adicional al tiempo que se disponga para realizar el entrenamiento de la máquina y las respuesta del mismo en el aprendizaje (Azure, 2019) . Teniendo esto presente, se tiene que hay ciertos parámetros para seleccionar el algoritmo que permita una clasificación óptima del grupo de datos estos son:

- Precisión.
- Tiempo de entrenamiento.

- Linealidad.
- Cantidad de parámetros.
- Cantidad de características.

7.3.2. Velocidad de respuesta en las técnicas de aprendizaje

Una vez el algoritmo de aprendizaje realiza la clasificación del grupo de canciones que ingresan se genera el modelo y el proceso en R permite saber el tiempo de procesamiento que tarda el algoritmo en clasificar y generar un nuevo modelo con el grupo que ingresa. Esto teniendo en cuenta la cantidad de veces que se repite el análisis de los datos mediante la función control y la variación de parámetros como la k en k -NN, el factor de corrección de Laplace (FL) y el ajuste en el clasificador bayesiano ingenio y el costo (C) o el sigma en el clasificador SVM. como se puede observar acontinuación donde se pueden observar los diferentes algoritmos de aprendizaje.

```

1 # Copyright (C) Ingenieria Electronica. 2018 UCundinamarca
2 #
3 # Algoritmo k vecinos mas cercanos
4
5 # Author: Jose Zamora <learningtesis@hotmail.com>
6 # Created: 2018
7
8
9 library(ggplot2)
10 library(dplyr)
11 library(tidyr)
12 library(purrr)
13 library(readr)
14 library(ISLR)
15 library(knitr)
16 library(tidyr)
17 library(kknn)
18 library(BBmisc)
19 library(MASS)
20 library(e1071)
21 library(caret)
22 library(AppliedPredictiveModeling)
23
24 generos <- read.csv("modelo.csv")
25 canciones <- read.csv("canciones.csv")
26

```

```

27 generos <- normalize(generos, method = "range", range = c(0, 1), margin = 1
   L, on.constant = "quiet")
28 canciones <- normalize(canciones, method = "range", range = c(0, 1), margin
   = 1L, on.constant = "quiet")
29 lista <- subset(canciones, select = -nombre)
30
31 # 1 grupo
32 train <- subset(generos, select = -nombre)
33 trainx <- train[,1:9]
34 trainy <- train[,10]
35
36 teste <- rbind(canciones[1:98,])
37 teste <- rbind(teste, canciones[8528:8625,])
38 teste <- rbind(teste, canciones[17054:17151,])
39 teste <- rbind(teste, canciones[25482:25579,])
40
41 testa <- rbind(lista[1:98,])
42 testa <- rbind(testa, lista[8528:8625,])
43 testa <- rbind(testa, lista[17054:17151,])
44 testa <- rbind(testa, lista[25482:25579,])
45
46 set.seed(392)
47 control <- trainControl(method="repeatedcv", number=10, repeats=3)
48 modelo <- train(Gene~., data=train, method="knn", trControl=control,
   preProcess= c("center", "scale"), tuneLength=20)
49 test <- testa
50 testx <- test[,1:9]
51 testy <- test[,10]
52 predic <- predict(modelo, testx)
53 prop.table(table(predic))*100
54 confusionMatrix(predic, testy)
55
56 generos <- rbind(generos, teste)
57 testa <- 0
58 teste <- 0

```

Listing 7: Algoritmo de aprendizaje k-NN

```

1 # Copyright (C) Ingenieria Electronica. 2018 UCundinamarca
2 #
3 # Algoritmo Clasificador bayesiano ingenuo
4
5 # Author: Jose Zamora <learningtesis@hotmail.com>
6 # Created: 2018
7
8

```

```

9 library(ggplot2)
10 library(dplyr)
11 library(tidyr)
12 library(purrr)
13 library(readr)
14 library(ISLR)
15 library(knitr)
16 library(tidyr)
17 library(kknn)
18 library(BBmisc)
19 library(MASS)
20 library(e1071)
21 library(caret)
22 library(klaR)
23
24 generos <- read.csv("modelo.csv")
25 canciones <- read.csv("canciones.csv")
26
27 generos <- normalize(generos, method = "range", range = c(0, 1), margin = 1
  L, on.constant = "quiet")
28 canciones <- normalize(canciones, method = "range", range = c(0, 1), margin
  = 1L, on.constant = "quiet")
29 lista <- subset(canciones, select = -nombre)
30 grid <- data.frame(fL=c(0,0.5,1.0), usekernel = TRUE, adjust=c(0,0.5,1.0))
31 # 1 grupo
32 train <- subset(generos, select = -nombre)
33 trainx <- train[,1:9]
34 trainy <- train[,10]
35
36 teste <- rbind(canciones[1:98,])
37 teste <- rbind(teste, canciones[8528:8625,])
38 teste <- rbind(teste, canciones[17054:17151,])
39 teste <- rbind(teste, canciones[25482:25579,])
40
41 testa <- rbind(lista[1:98,])
42 testa <- rbind(testa, lista[8528:8625,])
43 testa <- rbind(testa, lista[17054:17151,])
44 testa <- rbind(testa, lista[25482:25579,])
45
46 set.seed(392)
47 control <- trainControl(method="repeatedcv", number=10, repeats=3)
48 modelo <- train(Gene~., data=train, method="nb", trControl=control,
  preProcess= c("center", "scale"), tuneGrid = grid)
49 test <- testa
50 testx <- test[,1:9]
51 testy <- test[,10]

```

```

52 predic <- predict(modelo, testx)
53 prop.table(table(predic))*100
54 confusionMatrix(predic, testy)
55
56 generos <- rbind(generos, teste)
57 testa <- 0
58 teste <- 0

```

Listing 8: Algoritmo de aprendizaje Clasificador Bayesiano ingenuo

```

1 # Copyright (C) Ingenieria Electronica. 2018 UCundinamarca
2 #
3 # Algoritmo vector soporte de maquina
4
5 # Author: Jose Zamora <learningtesis@hotmail.com>
6 # Created: 2018
7
8 library(ggplot2)
9 library(dplyr)
10 library(tidyr)
11 library(purrr)
12 library(readr)
13 library(ISLR)
14 library(knitr)
15 library(tidyr)
16 library(kknn)
17 library(BBmisc)
18 library(MASS)
19 library(e1071)
20 library(caret)
21 library(kernlab)
22
23 generos <- read.csv("modelo.csv")
24 canciones <- read.csv("canciones.csv")
25
26 generos <- normalize(generos, method = "range", range = c(0, 1), margin = 1
  L, on.constant = "quiet")
27 canciones <- normalize(canciones, method = "range", range = c(0, 1), margin
  = 1L, on.constant = "quiet")
28 lista <- subset(canciones, select = -nombre)
29 grid <- expand.grid(sigma = c(.01, .015, 0.2), C = c(0.75, 0.9, 1, 1.1,
  1.25))
30
31 # 1 grupo
32 train <- subset(generos, select = -nombre)
33 trainx <- train[,1:9]

```

```

34 trainy <- train[,10]
35
36 teste <- rbind(canciones[1:98,])
37 teste <- rbind(teste, canciones[8528:8625,])
38 teste <- rbind(teste, canciones[17054:17151,])
39 teste <- rbind(teste, canciones[25482:25579,])
40
41 testa <- rbind(lista[1:98,])
42 testa <- rbind(testa, lista[8528:8625,])
43 testa <- rbind(testa, lista[17054:17151,])
44 testa <- rbind(testa, lista[25482:25579,])
45
46 set.seed(392)
47 control <- trainControl(method="repeatedcv", number=10, repeats=3)
48 modelo <- train(Gene~., data=train, method="svmRadial", trControl=control,
  preprocess= c("center", "scale"), tuneGrid=grid, tuneLength=10)
49 test <- testa
50 testx <- test[,1:9]
51 testy <- test[,10]
52 predic <- predict(modelo, testx)
53 prop.table(table(predic))*100
54 confusionMatrix(predic, testy)
55
56 generos <- rbind(generos, teste)
57 testa <- 0
58 teste <- 0

```

Listing 9: Algoritmo de aprendizaje SVM

Fragmento algoritmos realización gráficas: De acuerdo con los algoritmos de aprendizaje se realizan las gráficas de los modelos y los datos mediante gráficos de dispersión o pairs como se llama en R en 2D y 3D respectivamente en formato PDF.

```

1 # Copyright (C) Ingenieria Electronica. 2018 UCundinamarca
2 #
3 # Algoritmo con el cual se realizan las graficas de los modelos
4
5 # Author: Jose Zamora <learningtesis@hotmail.com>
6 # Created: 2018
7
8
9 library(ggplot2)
10 library(dplyr)

```

```

11 library(tidyr)
12 library(purrr)
13 library(readr)
14 library(ISLR)
15 library(knitr)
16 library(tidyr)
17 library(kknn)
18 library(BBmisc)
19 library(MASS)
20 library(e1071)
21 library(caret)
22 library(kernlab)
23 library(scatterplot3d)
24 library(plotly)
25
26 train$Gene[which(train$Gene == 0)] <- 'bambuco'
27 train$Gene[which(train$Gene == 1)] <- 'currulao'
28 train$Gene[which(train$Gene == 2)] <- 'joropo'
29 train$Gene[which(train$Gene == 3)] <- 'vallenato'
30 train$Gene <- as.factor(train$Gene)
31
32 p <- plot_ly(train, x = ~Ft, y = ~Ct, z = ~Ae, color = ~Gene, colors = c('
    red', 'blue', 'darkgreen', 'black')) %>%
33   add_markers() %>%
34   layout(scene = list(xaxis = list(title = 'Flujo espectral'),
35     yaxis = list(title = 'Centroide'),
36     zaxis = list(title = 'Atenuacion'))))
37
38 test$Gene[which(test$Gene == 0)] <- 'bambuco'
39 test$Gene[which(test$Gene == 1)] <- 'currulao'
40 test$Gene[which(test$Gene == 2)] <- 'joropo'
41 test$Gene[which(test$Gene == 3)] <- 'vallenato'
42 test$Gene <- as.factor(train$Gene)
43 t <- plot_ly(test, x = ~MFCCs4, y = ~MFCCs, z = ~Zt, color = ~Gene, colors
    = c('#BF382A', '#0C4B8E')) %>%
44   add_markers() %>%
45   layout(scene = list(xaxis = list(title = 'MFCCs4'),
46     yaxis = list(title = 'MFCCs'),
47     zaxis = list(title = 'Cruce por cero'))))
48
49 pdf("generos.pdf")
50 featurePlot(x = testx[1:4],
51   y = testy,
52   plot = "pairs",
53   ## Add a key at the top
54   auto.key = list(columns = 3))

```



```

55 featurePlot(x = testx[1:4],
56             y = testy,
57             plot = "ellipse",
58             ## Add a key at the top
59             auto.key = list(columns = 3))
60 featurePlot(x = testx[5:9],
61             y = testy,
62             plot = "pairs",
63             ## Add a key at the top
64             auto.key = list(columns = 3))
65 featurePlot(x = testx[5:9],
66             y = testy,
67             plot = "ellipse",
68             ## Add a key at the top
69             auto.key = list(columns = 3))
70 dev.off()
71 pdf("modelo.pdf")
72 plot(modelo)
73 featurePlot(x = trainx[1:4],
74             y = trainy,
75             plot = "pairs",
76             ## Add a key at the top
77             auto.key = list(columns = 3))
78 featurePlot(x = trainx[1:4],
79             y = trainy,
80             plot = "ellipse",
81             ## Add a key at the top
82             auto.key = list(columns = 3))
83 featurePlot(x = trainx[5:9],
84             y = trainy,
85             plot = "pairs",
86             ## Add a key at the top
87             auto.key = list(columns = 3))
88 featurePlot(x = trainx[5:9],
89             y = trainy,
90             plot = "ellipse",
91             ## Add a key at the top
92             auto.key = list(columns = 3))
93 dev.off()

```

Listing 10: Fragmento del algoritmo con el cual se grafican los diferentes modelos

7.3.3. Verificación del proceso de clasificación

Para el proceso de verificación del algoritmo de aprendizaje automático se basa en la validación que presenta (López y Molina, 2017), donde, se busca generalizar el conjunto de datos en búsqueda de aquellas características que sean extrapolables de manera que se optimiza evitando características irrelevantes. De manera que el algoritmo se ajusta mediante estrategias de validación cruzada

Validación cruzada de K iteraciones: La validación cruzada es un procedimiento de remuestreo que se utiliza para evaluar modelos de aprendizaje automático en una muestra de datos limitada. El procedimiento tiene un único parámetro llamado k que se refiere al número de grupos en que se dividirá una muestra de datos dada. Como tal, el procedimiento a menudo se denomina validación cruzada k -fold. Cuando se elige un valor específico para k , se puede usar en lugar de en la referencia al modelo, como $k = 10$ que se convierte en una validación cruzada de 10 veces.

La validación cruzada se usa principalmente en el aprendizaje automático aplicado para estimar la habilidad de un modelo de aprendizaje automático en datos invisibles. Es decir, usar una muestra limitada para estimar cómo se espera que el modelo se desempeñe en general cuando se usa para hacer predicciones sobre los datos que no se usaron durante la capacitación del modelo. Es un método popular porque es fácil de entender y porque generalmente resulta en una estimación menos sesgada o menos optimista de la habilidad del modelo que otros métodos, como una simple división de entrenamiento/prueba.

En definitiva, se estima cómo de exacto es el modelo generado. El procedimiento como se indica en la figura 6, donde cada color representa una clase es el siguiente:

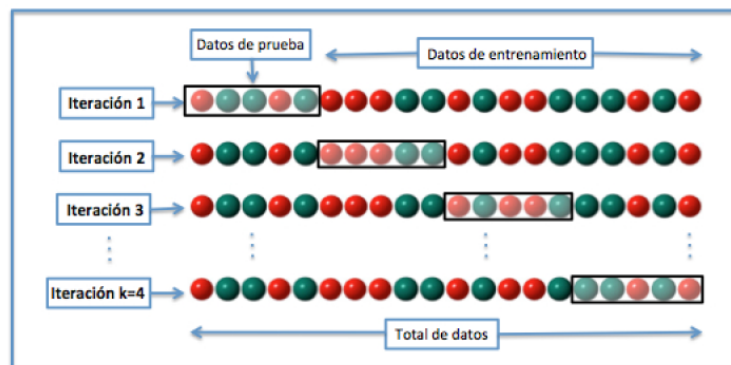


Figura 6: Validación cruzada de K iteraciones. Fuente: (López y Molina, 2017)

- El conjunto del que se parte se divide en K subconjuntos donde uno se asigna como el conjunto de prueba y la unión del resto ($K - 1$) se asignan como entrenamiento.
- El proceso de validación se repite K veces, de manera que todos los K subconjuntos hayan pasado por ser el conjunto de prueba.
- Una vez terminado el proceso, se calcula la media aritmética de las tasas de aciertos de cada iteración con su desviación típica estándar. Esto nos da una tasa de aciertos media con una desviación que podemos tomar como la exactitud aproximada que tiene el algoritmo con el conjunto de datos utilizado.

De manera que para realizar la validación se hacen grupos de 10 con 3 repeticiones, además evaluando que valor se ajustan mejor al modelo con la mejor sintonización del algoritmo; como se observa en las líneas de código donde la variable se llama `control`, y en el ajuste la función `train` donde se llama el `grid` de posibilidades de variación que se uso

7.4. Evaluar porcentaje precisión del algoritmo

Por último, para la evaluación de porcentaje de precisión del algoritmo se pretende generar un esquema con el porcentaje de precisión del software en la clasificación de los cuatro géneros folclóricos colombianos. Para este fin se aplica la matriz de similitud para determinar el porcentaje de precisión.

7.4.1. Matriz de confusión

Una matriz de confusión (Zelada, 2017), es una tabla que resume diferentes predicciones y resultados de pruebas y las compara con valores del mundo real. Las matrices de confusión se utilizan en estadísticas, minería de datos, modelos de aprendizaje automático y otras aplicaciones de inteligencia artificial. Una matriz de confusión también se puede llamar matriz de error.

Las matrices de confusión se utilizan para agilizar el análisis en profundidad de los datos estadísticos y facilitar la lectura de los resultados a través de una visualización clara de los datos. Las tablas pueden ayudar a analizar fallas en estadísticas, minería de datos, análisis forense y pruebas médicas. Un análisis exhaustivo ayuda a los usuarios a decidir qué resultados indican cómo se cometen los errores en lugar de simplemente evaluar el desempeño.

Las matrices de confusión utilizan un formato simple para registrar predicciones. En las filas de una matriz de confusión para un modelo de aprendizaje automático, las predicciones posibles se alinean en el lado derecho y las realidades en la parte superior. En las filas debajo de las realidades, se registran las predicciones o resultados. Los resultados pueden incluir la indicación correcta de un positivo como un positivo verdadero o un negativo como un negativo verdadero, así como un positivo incorrecto como un positivo falso o un negativo incorrecto como un falso negativo.

A continuación se muestra un ejemplo simple de una matriz de confusión que podría usarse para delinear los resultados.

	Clase 1 Predicción	Clase 2 Predicción
Clase 1 Actual	TP	FN
Clase 2 Actual	FP	TN

Tabla 7: Matriz de confusión

Las clases predichas están representadas en las columnas de la matriz, mientras que las clases reales están en las filas de la matriz. Entonces se tienen cuatro casos:

- Positivo (P): la observación es positiva (por ejemplo: es una manzana).
- Negativo (N): la observación no es positiva (por ejemplo: no es una manzana)
- Verdadero positivo (TP): la observación es positiva y se pronostica que será positiva.
- Falso negativo (FN): la observación es positiva, pero se pronostica negativa.
- Verdadero negativo (TN): la observación es negativa y se predice que es negativa
- Falso positivo (FP): la observación es negativa, pero se pronostica positiva.

Tasa de Clasificación/Precisión: La tasa de clasificación o precisión viene dada por la relación:

$$\text{Acierto} = \frac{TP + TN}{TP + TN + FP + FN} \quad (35)$$

Sin embargo, hay problemas con la precisión. Asume costos iguales para ambos tipos de errores. Una precisión del 99 % puede ser excelente, buena, mediocre, mala o terrible según el problema

Recordar: La memoria puede definirse como la relación entre el número total de ejemplos positivos clasificados correctamente y la división con el número total de ejemplos positivos. La recuperación alta indica que la clase se reconoce correctamente (número pequeño de FN). Viene dada por la relación:

$$\text{Recordar} = \frac{TP}{TP + FN} \quad (36)$$

Precisión: Para obtener el valor de la precisión, dividimos el número total de ejemplos positivos correctamente clasificados por el número total de ejemplos positivos previstos. Alta precisión indica que un ejemplo etiquetado como positivo es efectivamente positivo (pequeño número de PF). La precisión viene dada por la relación:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (37)$$

Alta memoria, baja precisión: Esto significa que la mayoría de los ejemplos positivos se reconocen correctamente (FN bajo) pero hay muchos falsos positivos.

Baja memoria, alta precisión: Esto muestra que faltamos muchos ejemplos positivos (FN alto), pero los que pronosticamos como positivos son ciertamente positivos (FP bajo)

Medida F: Dado que hay dos medidas (Precisión y memoria), es útil tener una medida que represente a ambas. Calcula una medida F que usa la media armónica en lugar de la media aritmética, ya que castiga más los valores extremos. La Medida F siempre estará más cerca del valor más pequeño de Precisión o Recuperación.

8. RESULTADOS

El desarrollo de la investigación y experimento en la caracterización y clasificación de la música colombiana en este trabajo esta enfocado en la determinación del algoritmo cuyos parámetros se ajusten en la detección de patrones para la clasificación de los géneros folclóricos; donde la recopilación musical se genero resultados tanto cualitativos como cuantitativos a pesar de la complejidad de las muestras musicales. En esta sección se muestran los resultados de comparación de los algoritmos en las diferentes etapas y la posterior selección del algoritmo de clasificación.

8.1. Estandarización de los cuatro géneros folclóricos colombianos

Teniendo en cuenta la estandarización para una aproximación inicial obtenida en las secciones 6.1.9 y 7.1.1; se procede a buscar en las plataformas digitales Spotify, Youtube; muestras musicales que cumplieran con estos parámetros básicos y validando los datos con un experto en música colombiana.

8.1.1. Audioteca

La adquisición de las muestras musicales de los cuatro géneros folclóricos se realiza mediante la siguiente líneas de comando.

```
1 # bambuco
2 youtube-dl --extract-audio --audio-format wav -o "%(title)s.%(ext)s" https
  ://www.youtube.com/watch?v=_C3HxQMCKs&list=
  PLai38kYR4FwFDidhZTXRw2QHHEd4S5_xG
3
4 # currulao
5 youtube-dl --extract-audio --audio-format wav -o "%(title)s.%(ext)s" https
  ://www.youtube.com/watch?v=B0KacydVgYo&list=PL1HQwCAKpq0SAYp5q_lb-YYe-
  B5nKfJPa
6
7 # joropo
8 youtube-dl --extract-audio --audio-format wav -o "%(title)s.%(ext)s" https
  ://www.youtube.com/playlist?list=PLBdzXBcUlXUQBhSMrSrvIN7nScVcnHlmG
9
10 # vallenato
11 youtube-dl --extract-audio --audio-format wav -o "%(title)s.%(ext)s" https
  ://www.youtube.com/playlist?list=PLFwnMbKJ9z9GGs63oAeP60A3oIBycy7yx
```

```
12 youtube-dl --extract-audio --audio-format wav -o "%(title)s.%(ext)s" https  
://www.youtube.com/playlist?list=PLGwRpZ40HjUsKsjAABhGS8apUU3RMGahx
```

Listing 11: Comandos configuración youtubedl

La salida de la línea de comando es la muestra musical en formato `.wav`, donde el nombre del archivo es el que se encuentra en el link en youtube.

En este proceso se busca obtener las muestras musicales con la mejor calidad posible teniendo presente que varias de las grabaciones son de disco LP y en la digitalización presentan un gran cantidad de ruido, se descartan estas pistas para de esta manera obtener los mejores resultados. Adicional se tiene presente que los géneros como el Bambuco y el Currulao, cuentan con representantes modernos donde las grabaciones no son de calidad debido a que son estudiantinas y autores emergentes, por otro lado, los géneros Joropo y Vallenato tiene muestras de audio de mayor calidad lo cual se ve en los resultados de la clasificación.

8.1.2. Banco de datos

En la recuperación de las muestras musicales se tiene en cuenta el trabajo realizado por (Muñoz, 2016), donde se hace uso y modificación de los algoritmos como se observa en la sección 7.1.3, de manera que se obtiene un archivo de formato `.csv` donde se almacenan las características de atenuación, centroide, flujo, cruce por cero y las primeras cinco características de coeficiente espectral en la frecuencia de Mel (MFCC); a continuación se observa la cabecera de la tabla de datos Figura 7 de características y algunos nombres de las canciones entrenadas y clasificadas en las siguientes tablas.

```

> head(canciones)
      Ct      Ae      Ft      Zt      MFCCs      MFCCs1      MFCCs2
1 0.9862905 0.03830226 0.00000000 0.08936170 0.4615967 0.4587144 0.3611811
2 0.9667747 0.03874810 0.02890026 0.10425532 0.4451825 0.4295935 0.3225711
3 0.9968596 0.03421116 0.03010957 0.09361702 0.4320869 0.4335518 0.3477754
4 0.9886029 0.03073172 0.04526750 0.09148936 0.4321849 0.4098125 0.3462525
5 0.9573358 0.02788323 0.02743184 0.09148936 0.4372307 0.4676112 0.4397704
6 0.9841142 0.02790153 0.04503379 0.09148936 0.4343285 0.5456323 0.4809987
      MFCCs3      MFCCs4
1 0.3697776 0.7267779
2 0.3465709 0.7541735
3 0.3225603 0.6995635
4 0.3459948 0.5446562
5 0.3242950 0.5270718
6 0.3704429 0.5845768

      nombre
1 /home/jestebanza/Aprendizaje/audicion computacional/canciones/bambuco/Alicia Isabel Santacruz - El Chambú.wav
2 /home/jestebanza/Aprendizaje/audicion computacional/canciones/bambuco/Alicia Isabel Santacruz - El Chambú.wav
3 /home/jestebanza/Aprendizaje/audicion computacional/canciones/bambuco/Alicia Isabel Santacruz - El Chambú.wav
4 /home/jestebanza/Aprendizaje/audicion computacional/canciones/bambuco/Alicia Isabel Santacruz - El Chambú.wav
5 /home/jestebanza/Aprendizaje/audicion computacional/canciones/bambuco/Alicia Isabel Santacruz - El Chambú.wav
6 /home/jestebanza/Aprendizaje/audicion computacional/canciones/bambuco/Alicia Isabel Santacruz - El Chambú.wav
      Gene
1 bambuco
2 bambuco
3 bambuco
4 bambuco
5 bambuco
6 bambuco

```

Figura 7: Cabecera tabla muestras musicales

Titulo	Artista	Genero
El chambú	Alicia Isabel Santacruz	Bambuco
Esta es mi tierra	Beatriz Arellano	Bambuco
Ausencia	Berenice Chávez	Bambuco
Canto fatal	Berenice Chávez	Bambuco
Morena	Berenice Chávez	Bambuco
El trapiche	Carlos Julio Ramírez	Bambuco
No sé por qué te quiero	Carlos Julio Ramírez	Bambuco
Mis dos amores	Carlos Julio Ramírez	Bambuco
Fusagasugueño	Conjunto Granadino - Pedro Morales Pino	Bambuco
El chambú	Coros Campesinos De Providencia Nariño	Bambuco
El trapiche	Coros Campesinos De Providencia Nariño	Bambuco
Bambuquisimo	Cuatro Palos	Bambuco
San Pedro en el Espinal	Cuerdas Colombianas	Bambuco
Noche que la vi tan negra	David Puerta	Bambuco
Yo era un niño	Diver Higuita	Bambuco

Tabla 8: Nombres muestras musical Bambuco entrenamiento

Titulo	Artista	Genero
Los camarones	Inés Granja	Currulao
A tumaco lo quemaron	Curupira	Currulao
A mi tierra	Interprete desconocido	Currulao
Adiós margarita	Grupo Canalón	Currulao
El eterno Dios	Agrupación Changó	Currulao
Yo quiero un hombre	Agrupación Tamafri	Currulao
Flolclor de mi tierra	Agrupación Changó	Currulao
Huele a currulao	Agrupación Changó	Currulao
La muluta	Agrupación Changó	Currulao
La pasada	Agrupación Changó	Currulao
Legado ancestral	Agrupación Changó	Currulao
Mi borojó	Agrupación Changó	Currulao
Mi peregoyo	Agrupación Changó	Currulao
Oilo guapi	Agrupación Changó	Currulao
Se oye sona	Agrupación Changó	Currulao

Tabla 9: Nombres muestras musical Currulao entrenamiento

Titulo	Artista	Genero
A paso lento	Aries Vigoth	Joropo
Señora soledad	Alí Cabello	Joropo
Sin tí muero	Alí Cabello	Joropo
Amor de peluche	Reynaldo Armas	Joropo
Agonia	Aries Vigoth	Joropo
El sida	Aries Vigoth	Joropo
Vivo tomando aguardiente	Armando Martines	Joropo
Baile de joropo	Biruaca de Oro	Joropo
Si el cielo es un paraiso	Cholo Valderrama	Joropo
Carmentea	Reynaldo Armas	Joropo
Sentimiento llanero	Carrao de Palmarito	Joropo
Mañana me voy pa pore	Cholo Valderrama	Joropo
Cuando el último llanero	Cholo Valderrama	Joropo
Muchacha cuanto te quiero	Cholo Valderrama	Joropo
El colector	Cholo Valderrama	Joropo

Tabla 10: Nombres muestras musical Joropo entrenamiento

Titulo	Artista	Genero
A pesar de todo	La Combinación Vallenata	Vallenato
Alguien como tu	Luis Mateus	Vallenato
Antes que te olvide	Dinastía Romero	Vallenato
Antojos	Los Chiches Del Vallenato	Vallenato
Ay hombre	Jorge Celedón	Vallenato
Busca un confidente	Los Diablitos	Vallenato
Como olvidarte	Binomio De Oro De América	Vallenato
Como expresar lo que siento	Binomio De Oro De América	Vallenato
Como hago	La Combinación Vallenata	Vallenato
Como te olvido	Binomio De Oro De América	Vallenato
Compréndeme	Iván Villazón	Vallenato
Corazoncito negro	Binomio De Oro De América	Vallenato
Corazón	El Clan Vallenato	Vallenato
Dame otra oportunidad	Jorge Celedón	Vallenato
De niña a mujer	Los Chiches Del Vallenato	Vallenato

Tabla 11: Nombres muestras musical Vallenato entrenamiento

Las observaciones de las características y nombres de las muestras musicales se dejan en el repositorio en formato `.csv` para facilidad del manejo en el documento.

De acuerdo con la sección 7.1, se obtiene la base de datos mediante la optimización de los códigos que presenta (Muñoz, 2016), donde se hace uso de la función `glob` con la cual se obtiene el listado de las canciones siendo una mejora al código para la extracción y generación de 98 observaciones por muestra musical de las nueve características las cuales se guardan en un archivo en formato `.csv`, de manera que se puede generar la estandarización de los cuatro géneros folclóricos mediante el análisis de las características dando una observación inicial a los datos donde se busca un cambio en los géneros para una posible separación lineal de manera manual, obteniendo que los coeficiente de Mel es un buen punto de partida para la separación e identificación de patrones en la clasificación de las muestras musicales. En otras palabras esta separación de los coeficientes de manera manual permite tener una primera elección en los algoritmos de clasificación que ejecutan la elección mediante agrupación o seccionamiento de los datos esto basado en la tabla 6.

8.2. Identificación de los métodos de aprendizaje

Teniendo presente la información recopilada en la sección 6.2.1, se realiza la extracción de la tabla 6, donde se analizan las diferentes características en los métodos de aprendizaje; a continuación se observa el proceso de selección de los tres métodos de aprendizaje seleccionados:

8.2.1. Esquema comparativo métodos de aprendizaje

Conforme el concepto de (Mathworks, 2018), se entiende que los algoritmos de aprendizaje automático, se diseñan para el reconocimiento de patrones naturales en los datos, los cuales generan conocimiento y contribuyen a la toma de una decisión acertada por parte del algoritmo, teniendo en cuenta que en la clasificación musical se presenta una gran cantidad de datos y variables donde la formulación de ecuaciones para la clasificación se vuelve compleja, se emplea diferentes técnicas de aprendizaje mediante supervisión del método o ejecutando algoritmos no supervisados, los cuales realizan la tarea.

De acuerdo con esto, la elección de los algoritmos de aprendizaje es una tarea compleja debido a las variables que intervienen en la clasificación y en donde los expertos en aprendizaje automático no pueden saber si un algoritmo funciona hasta probarlo. De manera que se tiene un concepto inicial en que ninguna de las técnicas de aprendizaje es mejor que la otra, teniendo presente esto no se puede aplicar de forma universal un método, por

lo cual para encontrar el algoritmo adecuado se utiliza la técnica de ensayo y error y se tiene en cuenta que pertenezca a la técnica supervisada de clasificación.

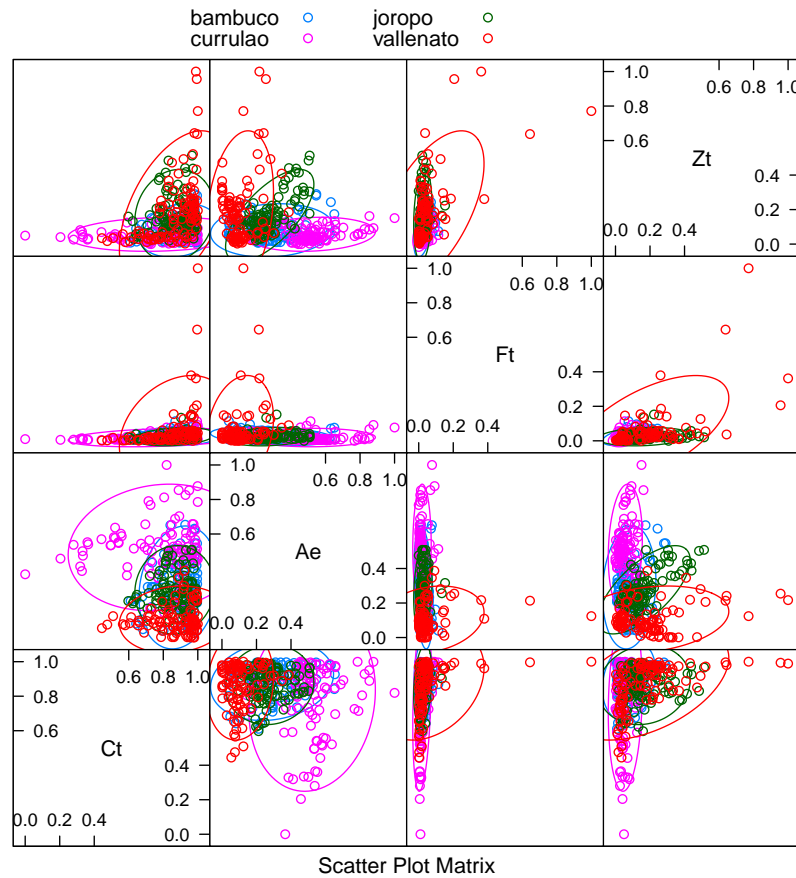


Figura 8: Gráfico de dispersión características

De manera que en la Figura 8, se puede observar cierta dificultad en la separación lineal de los datos debido a que las características de flujo espectral tienen su media de datos cerca a cero, la clasificación en cuanto a esa variable es complicada y se realiza el análisis de los datos mediante modelos de separación no lineal o modelos estadísticos como el clasificador bayesiano ingenuo o SVM con kernel variable buscando el mejor de estos, teniendo presente los datos de ingreso al modelo.

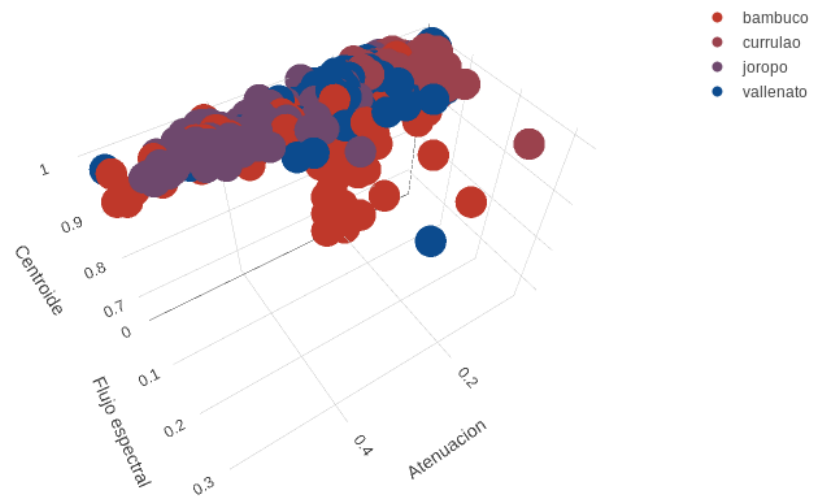


Figura 9: Características para clasificación en agrupación 3D

Teniendo presente la Figura 8, al no observarse facilidad en la separación lineal de los datos se opta por realizar una agrupación de tres características como en la Figura 9, de manera que se puede observar una separación de los datos de manera lineal, igualmente se observa que un algoritmo de clasificación SVM con kernel lineal no presenta una clasificación óptima, debido a que trabaja con duplas de características donde la separación lineal es menos compleja.

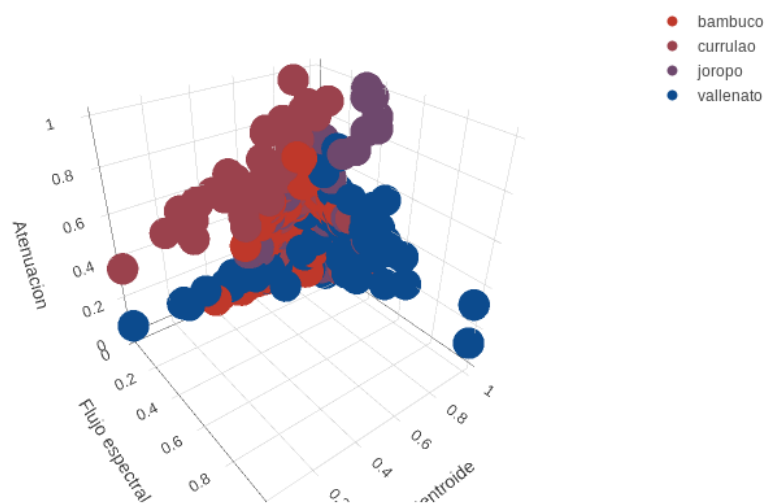


Figura 10: Características para clasificación en agrupación 3D

Como se observa en la Figura 9 y Figura 10, la clasificación se puede realizar mediante un algoritmo SVM con kernel radial debido a que permite una agrupación en diferentes características con datos donde la separación lineal no es posible realizar y donde se encuentran gran cantidad de datos y variables como sucede en la clasificación musical.

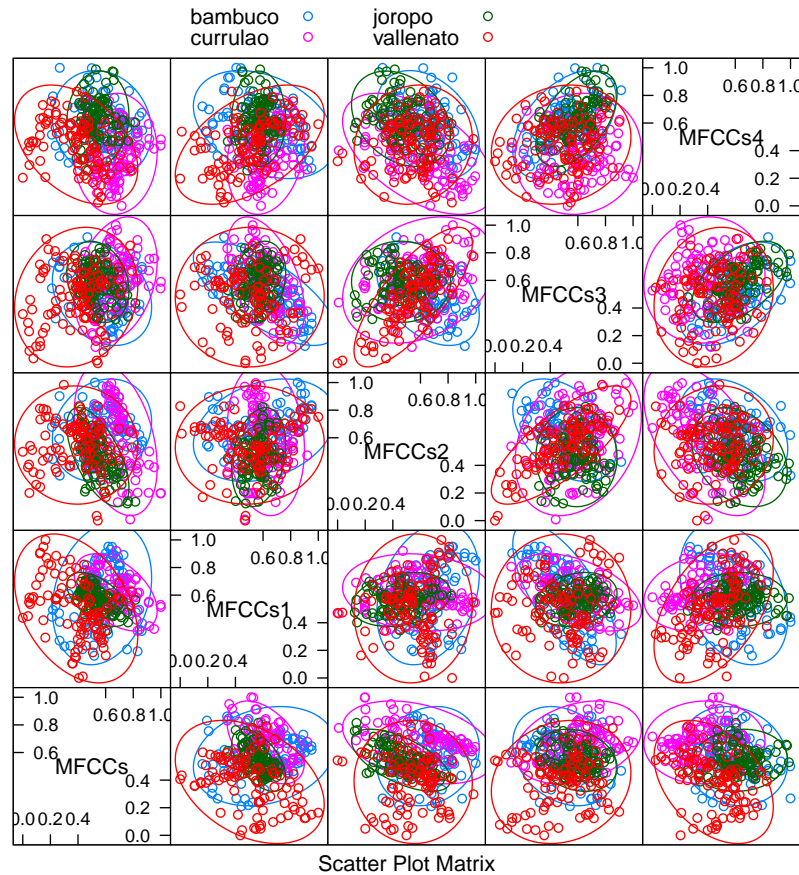


Figura 11: Gráfico de dispersión características MFCC

Como se observa en la Figura 11 las características MFCC presentan una mayor separación respecto a las observaciones en la Figura 8, por lo cual se tienen en cuenta las características de clasificación grupal presentada en k -NN donde se agrupa por distancia del parámetro k a los vecinos más cercanos en el proceso de clasificación, adicionalmente se observa que la media de los datos en las características se encuentra en 0.5 y la separación es moderada como se puede distinguir en las elipses de agrupación permitiendo tener una claridad en la selección de los métodos de aprendizaje.

8.2.2. Comparación métodos de aprendizaje

Conforme a las características de los métodos de aprendizaje se comprende que son tres los métodos que se ajustan a la clasificación deseada. Siendo k -NN el método con el que se parte para realizar la enseñanza, debido a la velocidad de entrenamiento y la poca ajustabilidad del modelo final; seguido se realiza la enseñanza del clasificador bayesiano ingenuo, el cual al ser un modelo estadístico y de análisis de series de datos permite mediante el cambio de la curva gaussiana en el tiempo, iniciar la búsqueda de patrones en los datos. Por último, debido a la no linealidad de los datos se profundiza en el método SVM con kernel radial en la enseñanza y generación de un modelo de clasificación con parámetros de enseñanza y validación variable teniendo los siguientes modelos finales.

El proceso de entrenamiento de los tres algoritmos se realiza seleccionando las quince primeras canciones de cada género observadas en las tablas de la sección 8.1.2, esto por igual para los tres métodos de aprendizaje obteniendo las siguientes figuras; donde se puede observar las curvas de aprendizaje de los algoritmos por genero folclórico.

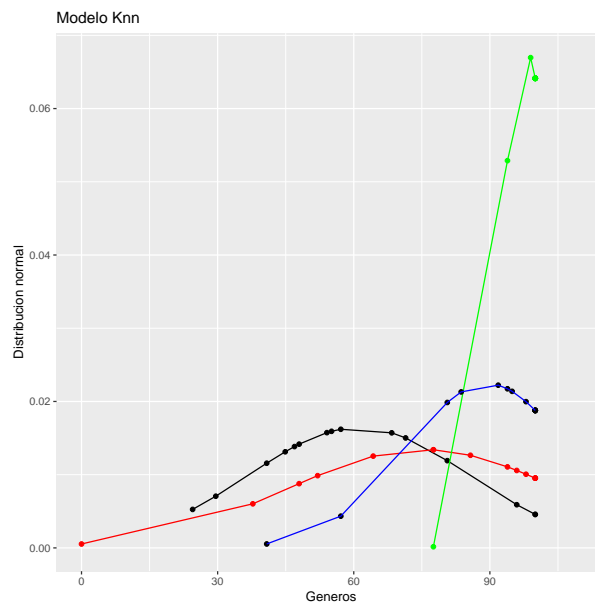


Figura 12: Curva de aprendizaje modelo Knn.

Teniendo presente las curvas gaussianas del algoritmo de aprendizaje de la Figura 12, se observa un comportamiento de clasificación similar en cuanto a los géneros, siendo la curva gaussiana representada por el color negro correspondiente al bambuco con una media de clasificación del 64 % y una desviación de los datos de 0.01 en promedio seguido por el currulao identificado con la curva gaussiana roja con una media de aprendizaje de 84 % y una desviación del 0.015 teniendo una variación mayor en el porcentaje de clasificación;

por otro lado, el joropo identificado con la curva gaussiana verde presenta una media en el aprendizaje de clasificación superior al 96 % con una desviación del 0.05 y por ultimo el vallenato identificado con la curva gaussiana azul tiene una media de aprendizaje del 90 % y una desviación estándar del 0.02. Permite obtener al modelo k -NN una media de clasificación en los cuatro géneros del 86 % siendo un modelo con un alto grado de exactitud para la clasificación de los géneros folclóricos.

De manera que en el análisis de las curvas gaussianas en el modelo k -NN se tiene una variación en el porcentaje de acierto mínimo, por lo tanto las curvas son menos altas y con un ajuste de la media de los datos moderada en el caso del bambuco y el currulao donde se observa una curva gaussiana clara con un desfase en la media de clasificación de 20 %, por otro lado las curvas de joropo y vallenato presentan una mayor variación en los porcentajes, sin embargo, su media de clasificación se encuentra por encima del 90 % con un desfase del 4 %, teniendo un porcentaje de clasificación muy bueno y valido para considerar en la selección de la biblioteca donde las muestras musicales son grabaciones de calidad.

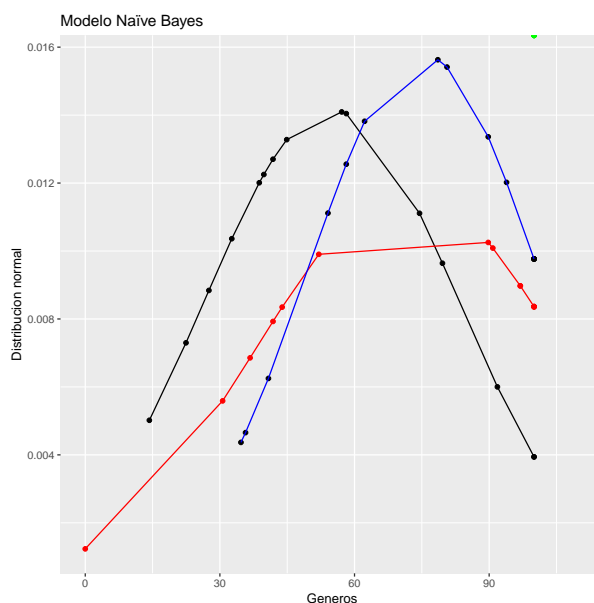


Figura 13: Curva de aprendizaje algoritmo clasificador bayesiano ingenuo.

Ciertamente el algoritmo clasificador bayesiano ingenuo, no esta diseñado para la clasificación cuantitativa. Como se puede observar en la Figura 13 son las observaciones en la clasificación de los cuatro géneros folclóricos, sin embargo, al ser un modelo que permite el reconocimiento de patrones de manera interactiva por el planteamiento de modelos estadísticos los cuales determinan los parámetros, se encontró que tiene una funcionalidad en la solución del algoritmo clasificador; aún siendo el algoritmo con la media de clasificación

más baja de los tres algoritmos con el 82 %, y siguiendo las convenciones de identificación de las curvas de aprendizaje del algoritmo k -NN, es un algoritmo el cual clasifica el bambuco con una media de 59 % y una desviación del 0.0010 de variación; el currulao con una media de clasificación del 82 % con una desviación de los datos 0.006; por otro lado el algoritmo bayesiano tiene una clasificación del joropo del 100 % sin desviación de los datos y por ultimo el vallenato con una media de clasificación del 87 % con una desviación del 0.008.

El clasificador bayesiano ingenuo presenta una variación en la desviación de los datos mayor a la presentada en el algoritmo k -NN, sin embargo, presenta curvas de aprendizaje con mayor claridad, con un desfase del 15 % en el porcentaje de acierto entre los géneros bambuco, currulao y vallenato y no presenta curva de aprendizaje del joropo debido a su 100 % de acierto en el entrenamiento del genero.

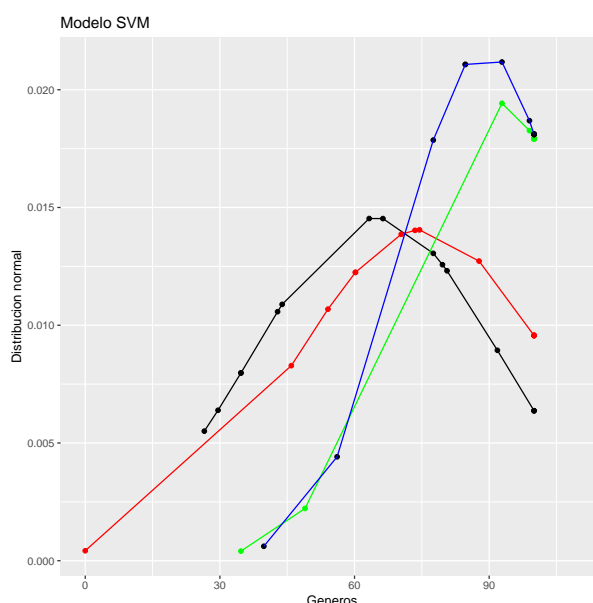


Figura 14: Curva de aprendizaje modelo SVM con kernel Radial.

A este análisis se añade, el aprendizaje realizado por el algoritmo SVM que se muestra en la Figura 14 con una media de acierto de clasificación del 89 %, al igual que los métodos anteriormente mencionados se realiza el entrenamiento de 15 piezas musicales por genero, obteniendo una media del 68 % para el bambuco con una desviación de los datos de 0.008, el currulao con un media del 89 % con una desviación del 0.0014, igualmente el algoritmo clasifica el joropo con una media del 100 % con una desviación del 0.0020 teniendo poca variación en los datos y el vallenato se entrena con una media del 94 % de acierto con una desviación en los datos de 0.0021.

El algoritmo SVM presenta un desfase entre el bambuco y el currulao del 20 % y entre el currulao y los otros géneros del 6 % y del 12 %, siendo el algoritmo que presenta un menor desplazamiento en el porcentaje de acierto en el entrenamiento, aunque presenta una variación en la desviación de la media de los datos se presta para catalogar con un alto grado de acierto el genero objetivo, por la complejidad del algoritmo con kernel radial permite obtener un clasificador con el 89 % de acierto en los cuatro géneros folclóricos.

Método de aprendizaje	Porcentaje de acierto
k -Vecinos más cercanos	86 %
Clasificador bayesiano ingenuo	82 %
Soporte vector de máquina con kernel radial	89 %

Tabla 12: Porcentaje de acierto técnica manual

De acuerdo con el análisis realizado en la sección 7.2, se obtiene una respuesta previa a la selección de los posibles métodos obteniendo algoritmos los cuales realizan la clasificación mediante agrupación por conjuntos como k -NN, un análisis estadístico realizando reconocimiento de patrones en las observaciones de las muestras musicales y por ultimo generando una aproximación en los datos mediante la identificación de los vectores soportes con los cuales el algoritmo realiza la clasificación, obteniendo la Tabla 12 en la cual se pueden observar los porcentajes de acierto realizado manualmente durante el entrenamiento de los tres algoritmos seleccionados, mediante la identificación de usabilidad, metodología y aplicabilidad al conjunto de datos obtenido de las muestras musicales.

De todo ello se obtienen tres algoritmos con una media de clasificación superior al 80 % lo cual para el entrenamiento de máquina y teniendo en cuenta las características y resultados que se buscaban obtener, se tiene un entendimiento de los géneros folclóricos bastante bueno, indicando que es posible catalogar los géneros mediante una medida cuantitativa la cual es extraída en la sección 8.1.

8.3. Técnica para la clasificación

De acuerdo con la metodología planteada para la selección del algoritmo de aprendizaje, se tiene presente la estimación de acierto en la sección 8.2.2 donde se realiza el procedimiento de manera manual tomando los valores de clasificación y realizando un promedio de los datos para obtener las figuras de la sección 8.2.2, por lo tanto se procede, mediante línea de comandos en el algoritmo a obtener el porcentaje de acierto en cuanto a clasificación de los

cuatro géneros, la velocidad con la cual realizo el entrenamiento y la obtención del modelo final mediante la validación cruzada de k iteraciones como se muestra a continuación.

8.3.1. Inspección en la clasificación para la definición de la técnica

De acuerdo con lo planteado por (Mathworks, 2018), se tuvieron en cuenta las características de selección del algoritmo propuesto en la sección 7.3.1, donde se analizan la precisión del algoritmo con cada grupo de aprendizaje, el tiempo de entrenamiento, la linealidad en el proceso de aprendizaje. Por otro lado se debe tener en cuenta el algoritmo con el cual una gran cantidad de datos y características de análisis tiene un menor grado de incertidumbre en el proceso de clasificación. Teniendo esto presente se procede a realizar el proceso de validación de estos parámetros en el algoritmo como se observa en las siguientes figuras.

Modelo k -NN: Como se puede observar en la Figura 15 y la Figura 16 comparando los grupos de aprendizaje en el modelo, el primer grupo presenta un entendimiento mayor de los datos con un porcentaje de acierto en los 5 vecinos más cercanos con el 84 % de clasificación de los géneros, y en el modelo final presenta un entendimiento del 75 % de los géneros con un porcentaje de acierto menor, sin embargo, se percibe un grado de incertidumbre menor en los datos debido a un grupo de enseñanza mayor, lo cual se puede observar en la curva final, donde se presenta una linealidad en la clasificación en la variación del parámetro k , al ser un algoritmo el cual realiza su clasificación mediante la medición de la distancia euclidiana entre la observaciones se tiene que con un grupo mayor de entrenamiento, con las mismas 9 características del vector de textura tímbrica el porcentaje de acierto tiende a bajar de acuerdo con las características del algoritmo k -NN.

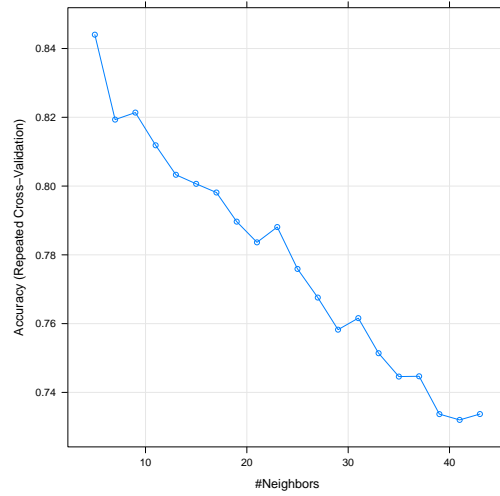


Figura 15: Modelo aprendizaje primer grupo.

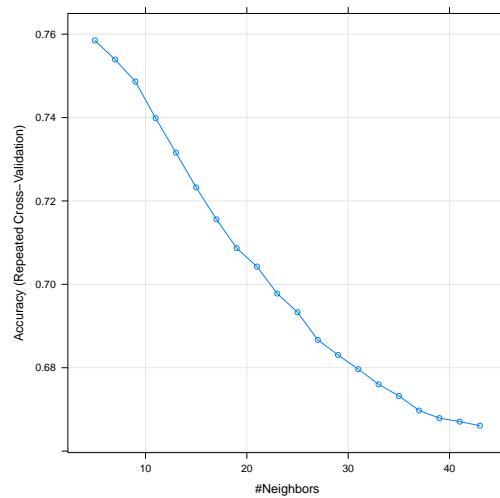


Figura 16: Modelo aprendizaje último grupo.

Adicionalmente debido a la complejidad de las muestras el algoritmo de aprendizaje tiene dificultad de clasificación con un parámetro k mayor a veinte debido a la cercanía en las observaciones. El algoritmo presenta un tiempo de entrenamiento total de 30 minutos, teniendo presente que son 6272 observaciones las cuales tiene el modelo final para la toma en la decisión, la diferencia de tiempo que tardó en realizar el primer grupo con una validación de los k parámetros y la triple validación realizada la cual se explicará en la sección 8.3.3.

Modelo clasificador bayesiano ingenuo: El proceso estadístico manejado por el clasificador bayesiano ingenuo permite realizar una corrección del factor de Laplace, el cual determina la distribución normal y genera los puntos de cambio para el ajuste del ancho de banda como se observa en la figura 17 y la Figura 18 la ganancia cambia con respecto a la variación del factor de Laplace, en el cual el modelo tiene el porcentaje de acierto en 78 % en el primer grupo de enseñanza, y de 51 % en el último grupo de aprendizaje, siendo el modelo menos eficiente al momento de clasificar los géneros.

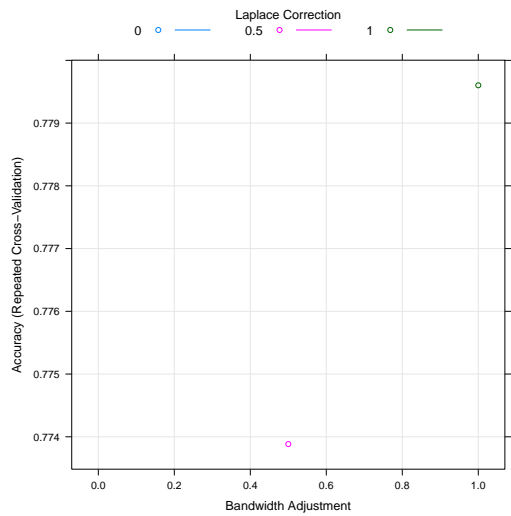


Figura 17: Modelo aprendizaje primer grupo.

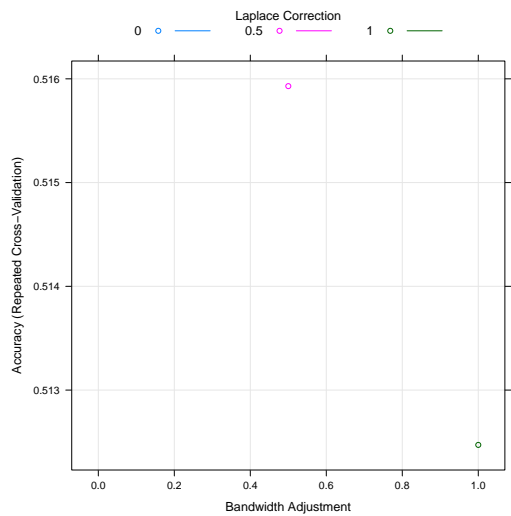


Figura 18: Modelo aprendizaje último grupo.

El clasificador bayesiano ingenuo es un algoritmo enfocado a la detección y reconocimiento de texto, sus características en la detección de patrones exalta la exactitud en el acierto

y clasificación de spam, sin embargo el algoritmo mantiene un buen desempeño de 51 % aún con el aprendizaje completo y con un tiempo de entrenamiento de 40 minutos desde el primer grupo hasta el último, se presenta como un buen algoritmo de validación en la clasificación musical como segunda opción, debido a la facilidad de aplicación en modelos estadísticos recibe grandes cantidades de datos y características permitiendo obtener un porcentaje de validación en clasificación medio, la linealidad del algoritmo es baja debido la formulación de las ecuaciones usadas por este, las cuales tienden a volver lineal las características sin tener en cuenta la función de las mismas en el dominio del tiempo o la frecuencia. De manera que al ser la segunda opción al momento de realizar la clasificación permite eliminar el factor de la linealización de los datos con solo la validación en la toma de decisiones del algoritmo.

Modelo SVM: El modelo SVM al ser el más complejo tiene un grado de acierto mayor en el proceso de aprendizaje como se observa en la Figura 19 y la Figura 21 en donde, el primer grupo presenta un aprendizaje del 89 %, al igual que en los métodos anteriores el grado de incertidumbre en el primer grupo es más grande, sin embargo con el último grupo de enseñanza presenta un 74 % de acierto, una de las desventajas es la capacidad de procesamiento en donde SVM tiene un análisis lento como se observó en la tabla 6.

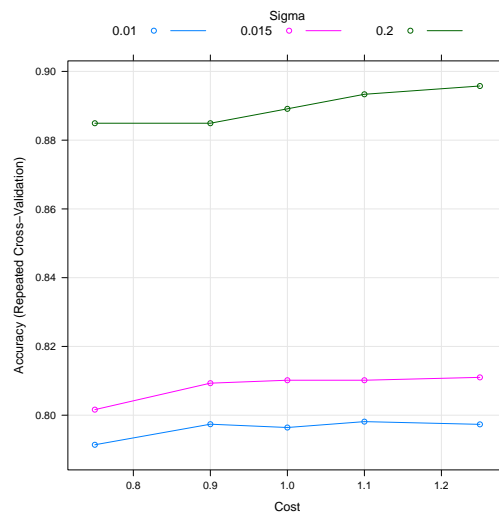


Figura 19: Modelo aprendizaje primer grupo.

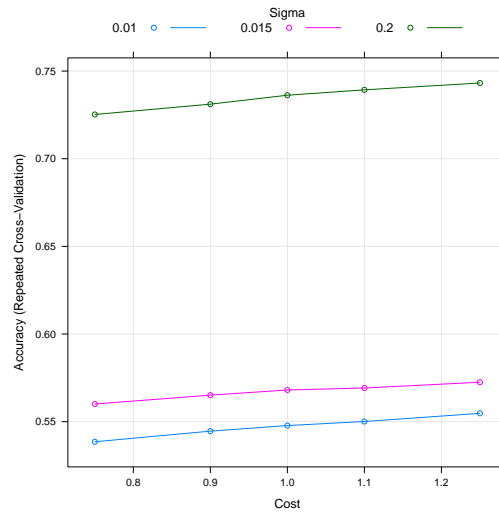


Figura 20: Modelo aprendizaje último grupo.

De acuerdo con las características del algoritmo SVM se presenta un porcentaje de acierto mayor debido a que trabaja con un kernel radial el cual encuentra agrupaciones en las observaciones cuyas características estén separadas, del mismo modo el algoritmo, presenta una formulación de aprendizaje no lineal y un proceso de de validación muy lento debido a la ejecución de sintonización donde tarda cerca de 2 horas desde el primer grupo hasta el último grupo de aprendizaje, a continuación el algoritmo tiene una interacción con gran cantidad de datos y características lo cual lo hace el algoritmo con los parámetros necesarios para realizar la clasificación de los géneros folclóricos.

Método de aprendizaje	Porcentaje de acierto manual	Porcentaje de acierto código	Media de acierto del algoritmo
k -Vecinos más cercanos	86 %	75 %	80,5 %
Clasificador bayesiano ingenuo	82 %	51 %	66,5 %
Soporte vector de máquina con kernel radial	89 %	74 %	81,5 %

Tabla 13: Porcentaje de acierto métodos

Realizando la inspección de los algoritmos y buscando la elección de un algoritmo se tiene que tener en cuenta la Tabla 13, donde el porcentaje de acierto obtenido en la sección 8.2.2 y comparando con lo obtenido en esta sección se tiene que el algoritmo SVM tiene un 81 % de acierto sobre el método k -NN con un 80 %, al tener en cuenta que el algoritmo

SVM está basado en la parametrización y agrupación que presenta k -NN se selecciona como la mejor opción aun siendo el algoritmo con un entrenamiento más lento, después de obtener el modelo o los vectores soporte, su clasificación la limita el equipo en el cual se ejecuta el algoritmo; en conclusión como se mencionó anteriormente para trabajos futuros se propone el clasificador bayesiano ingenuo como validador del algoritmo SVM.

8.3.2. Velocidad de respuesta en las técnicas de aprendizaje

Hay que tener en cuenta la rapidez que identifica a esta modalidad de inteligencia artificial, resulta de la potencia en el procesamiento, tanto secuencial como paralela y de la capacidad de almacenamiento en memoria. Sin embargo, el potencial del aprendizaje automático para mejorar otras técnicas de análisis de datos tradicionales, tiene que ver con la capacidad de volver a aprender de los datos. Con lo anterior se puede deducir que el punto fuerte de este tipo de métodos no es sólo velocidad.

La ajustabilidad de los datos no sólo permite que las soluciones predictivas basadas en sofisticados algoritmos sean más exactas, sino que también impulsan la importancia de la rapidez del software, de esta forma, es posible interpretar en tiempo real los miles de millones de filas y columnas que hay para investigar, a la vez que no se detiene el análisis del flujo de datos de ingreso.

Modelo k -NN: Teniendo presente los parámetros de ajustabilidad del modelo que se observa mediante el análisis en las características del entrenamiento, se obtienen las siguientes figuras donde se muestra el tiempo que tarda el algoritmo de aprendizaje en realizar la clasificación y ejecutar la muestra de los datos al usuario.

```
> modelo$times
$everything
  user  system elapsed
5.516   0.036   5.711

$final
  user  system elapsed
0.008   0.000   0.006
```

Figura 21: Velocidad aprendizaje primer grupo.

De manera que se obtiene una diferencia de ocho veces entre el lapso de muestra de los datos en el primer grupo de aprendizaje y el último grupo de aprendizaje en el modelo final como se observa en la Figura 21 y Figura 22.


```

> modelo$times
$everything
  user  system elapsed
94.132   0.048  95.245

$final
  user  system elapsed
 0.044   0.004   0.049

```

Figura 22: Velocidad aprendizaje último grupo.

Modelo Clasificador bayesiano ingenuo: De acuerdo a las características estadísticas se plantea un modelo inicial sencillo con un grupo de nueve características de observación donde se estiman: media, mediana, desviación estándar y cuartiles; esto para los cuatro géneros folclóricos, siendo un proceso que tiene los tiempos de análisis como los muestra en la Figura 23 y la Figura 24.

```

> modelo$times
$everything
  user  system elapsed
 7.616   0.040   7.697

$final
  user  system elapsed
 0.048   0.000   0.048

```

Figura 23: Velocidad aprendizaje primer grupo.

```

> modelo$times
$everything
  user  system elapsed
69.356   0.020  70.068

$final
  user  system elapsed
 0.096   0.000   0.095

```

Figura 24: Velocidad aprendizaje último grupo.

Se obtiene que el proceso de entrenamiento de una mayor cantidad de datos tiende a ser un factor al momento de generar el modelo final, debido a la complejidad en las tablas de análisis mencionadas anteriormente, por lo cual se demora en el proceso dos veces respecto a la velocidad del modelo inicial y siendo un modelo dos veces más lento que el modelo k -NN.

Modelo SVM: De acuerdo a la similitud en el proceso de organización y ajustabilidad del algoritmo SVM con el modelo k -NN se parte del concepto que el tiempo de entrenamiento se debe asimilar a valores cercanos, sin embargo, se presenta una diferencia en el proceso debido al algoritmo no lineal que se aplica, donde existe una diferencia de diez

veces entre la clasificación del modelo SVM y el modelo k -NN, adicionalmente se tiene la validación cruzada de los modelos y la variación de los vectores de ajustabilidad donde el modelo SVM presentó debilidad en la velocidad de procesamiento.

```
> modelo$time
$everything
  user  system elapsed
3.784  0.008  3.798

$final
  user  system elapsed
0.06   0.00   0.06
```

Figura 25: Velocidad aprendizaje primer grupo.

```
> modelo$times
$everything
  user  system elapsed
402.080 109.172 340.996

$final
  user  system elapsed
14.156  3.624 11.745
```

Figura 26: Velocidad aprendizaje último grupo.

En el análisis de respuesta del algoritmo en la clasificación de las muestras musicales se observa que el modelo SVM posee una velocidad de clasificación menor de acuerdo con un grupo de aprendizaje pequeño, sin embargo con el aprendizaje completo se observa en la Figura 25 y en la Figura 26 que el algoritmo que presenta una mayor velocidad es el método k -NN.

8.3.3. Verificación del proceso de clasificación

De acuerdo con la sección 7.3.3 se verifica la validación cruzada de k iteraciones planteada en (López y Molina, 2017), de manera que se obtienen las siguientes figuras y para los tres métodos de aprendizaje se aplica la validación cruzada de diez iteraciones con tres repeticiones por método para la verificación del aprendizaje de los datos.

Método k -NN: Como se observa en la Figura 27 y la Figura 28 se tiene una validación cruzada de diez iteraciones con tres repeticiones de los datos obteniendo un modelo con las características observadas para el ajuste del parámetro k del modelo.

```

k-Nearest Neighbors
392 samples
9 predictor
4 classes: 'bambuco', 'currulao', 'joropo', 'vallenato'

Pre-processing: centered (9), scaled (9)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 353, 352, 352, 353, 353, 353, ...
Resampling results across tuning parameters:

k  Accuracy  Kappa
5  0.8473831  0.7965899
7  0.8261276  0.7682256
9  0.8213765  0.7618976
11 0.8109827  0.7480196
13 0.8084423  0.7446294
15 0.8031934  0.7376756
17 0.7981743  0.7309657
19 0.7854514  0.7140725
21 0.7803469  0.7073109
23 0.7812656  0.7085329

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.

```

Figura 27: Validación cruzada 10 iteraciones grupo inicial

```

k-Nearest Neighbors
5880 samples
9 predictor
4 classes: 'bambuco', 'currulao', 'joropo', 'vallenato'

Pre-processing: centered (9), scaled (9)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 5292, 5292, 5292, 5292, 5292, 5292, ...
Resampling results across tuning parameters:

k  Accuracy  Kappa
5  0.7585034  0.6780045
7  0.7539116  0.6718821
9  0.7486395  0.6648526
11 0.7398526  0.6531368
13 0.7315760  0.6421013
15 0.7232426  0.6309902
17 0.7155896  0.6207861
19 0.7086735  0.6115646
21 0.7042517  0.6056689
23 0.6977891  0.5970522
25 0.6933107  0.5910809
27 0.6866780  0.5822373
29 0.6830499  0.5773998
31 0.6796485  0.5728647
33 0.6760204  0.5680272
35 0.6732426  0.5643235
37 0.6697279  0.5596372
39 0.6679138  0.5572184
41 0.6670635  0.5560847
43 0.6660998  0.5547997

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.

```

Figura 28: Validación cruzada 10 iteraciones grupo final

De acuerdo con la Figura 27 y la Figura 28 se obtiene un algoritmo de cuatro clases con nueve variables predictivas y un grupo de análisis de 5800 observaciones finales con las cuales se ejecuta el modelo, además se tiene un análisis de acierto para los valores de k donde se presento un porcentaje de acierto del 75 % en el modelo final.

Modelo clasificador bayesiano ingenuo: De acuerdo con los ajustes de parámetros para el kernel del clasificador bayesiano ingenuo se selecciona el valor óptimo donde se puede evaluar el modelo final de manera que se obtiene un factor de Laplace de 0.5 con un ajuste del modelo de 0.5 y teniendo presente la validación cruzada de 10 iteraciones con tres repeticiones del grupo de datos se obtienen la Figura 29 y la Figura 30.

```
Naive Bayes
392 samples
9 predictor
4 classes: 'bambuco', 'currulao', 'joropo', 'vallenato'

Pre-processing: centered (9), scaled (9)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 353, 352, 352, 353, 353, 353, ...
Resampling results across tuning parameters:

  fL  adjust  Accuracy  Kappa
0.0  0.0      NaN      NaN
0.2  0.2      0.6996299  0.5996438
0.5  0.5      0.7738841  0.6984968

Tuning parameter 'usekernel' was held constant at a value of TRUE
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0.5, usekernel = TRUE and
adjust = 0.5.
```

Figura 29: Validación cruzada 10 iteraciones grupo inicial

```
Naive Bayes
5880 samples
9 predictor
4 classes: 'bambuco', 'currulao', 'joropo', 'vallenato'

Pre-processing: centered (9), scaled (9)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 5292, 5292, 5292, 5292, 5292, ...
Resampling results across tuning parameters:

  fL  adjust  Accuracy  Kappa
0.0  0.0      NaN      NaN
0.2  0.2      0.5082200  0.3442933
0.5  0.5      0.5159297  0.3545729

Tuning parameter 'usekernel' was held constant at a value of TRUE
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0.5, usekernel = TRUE and
adjust = 0.5.
```

Figura 30: Validación cruzada 10 iteraciones grupo final

Con un modelo de 5880 observaciones se tiene que el modelo clasificador bayesiano ingenuo presenta 51 % de acierto en el proceso de clasificación con cuatro clases categóricas y nueve variables predictivas las cuales son validadas en el preprocesamiento de los datos.

Como se observó en la sección 8.3.1 el clasificador bayesiano ingenuo presento un porcentaje de acierto bajo en el entrenamiento pero se presenta como un algoritmo validador en la toma y selección del genero a clasificar, debido a las características estadísticas de clasificación y validación que propias del clasificador.

Modelo SVM: De acuerdo con el análisis en la validación cruzada de k iteraciones se observó en el entrenamiento del modelo SVM que este proceso de validación genera un tiempo de entrenamiento mayor respecto a los algoritmos anteriores, por lo cual se realiza un análisis de la Figura 21 para obtener un valor de costo y de sigma el cual permitiese obtener el modelo más rápido con la validación cruzada de diez iteraciones como se observa en la Figura 32.

```
Support Vector Machines with Radial Basis Function Kernel

392 samples
 9 predictor
 4 classes: 'bambuco', 'currulao', 'joropo', 'vallenato'

Pre-processing: centered (9), scaled (9)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 353, 352, 352, 353, 353, ...
Resampling results across tuning parameters:

sigma C      Accuracy  Kappa
0.01  0.10  0.6335049  0.5142394
0.01  0.20  0.7453938  0.6607555
0.01  0.50  0.7820124  0.7094103
0.01  1.00  0.7964186  0.7285671
0.01  1.25  0.7973386  0.7297859
0.02  0.10  0.7456178  0.6611218
0.02  0.20  0.7701286  0.6936525
0.02  0.50  0.8025141  0.7366881
0.02  1.00  0.8144689  0.7526324
0.02  1.25  0.8152382  0.7536808
0.05  0.10  0.7820989  0.7096030
0.05  0.20  0.8024927  0.7366897
0.05  0.50  0.8245330  0.7660604
0.05  1.00  0.8398368  0.7864871
0.05  1.25  0.8450348  0.7934262

Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.05 and C = 1.25.
```

Figura 31: Validación cruzada 10 iteraciones grupo inicial

```
Support Vector Machines with Radial Basis Function Kernel

6272 samples
 9 predictor
 4 classes: 'bambuco', 'currulao', 'joropo', 'vallenato'

Pre-processing: centered (9), scaled (9)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 5646, 5645, 5645, 5644, 5645, 5644, ...
Resampling results:

Accuracy  Kappa
0.7395908  0.652787

Tuning parameter 'sigma' was held constant at a value of 0.2
Tuning
parameter 'C' was held constant at a value of 1.25
```

Figura 32: Validación cruzada 10 iteraciones grupo final

En el proceso de validación del algoritmo SVM se obtienen diferentes características del modelo que permiten realizar la clasificación teniendo presente la Figura 33 y la Figura 34 se entiende que el aprendizaje resalta con una mayor cantidad de datos algunas características como el flujo espectral y el centroide con respecto de las otras en el vector de observaciones.

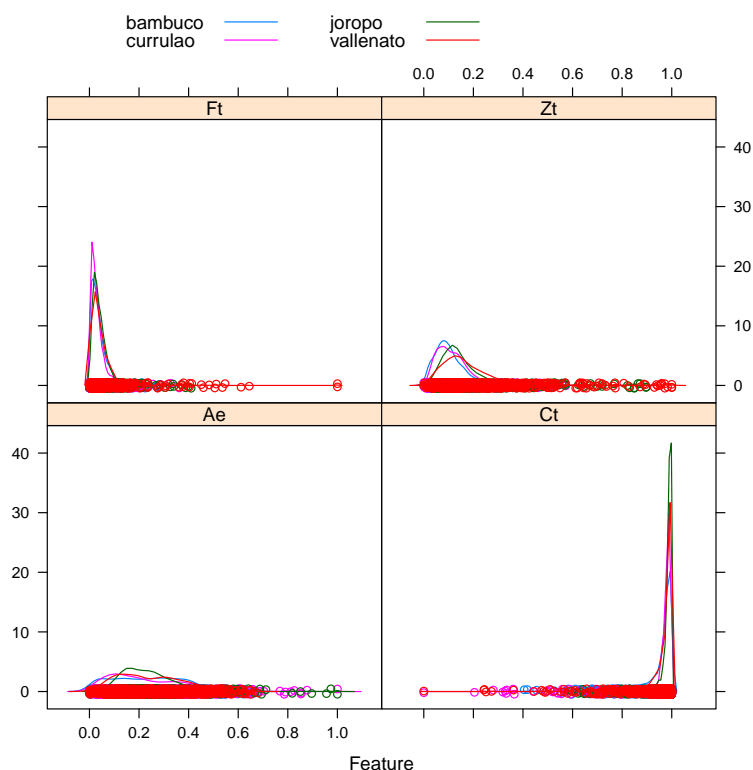


Figura 33: Densidad e identificación de características en el entrenamiento

Inicialmente se observa en la Figura 33 el realce que presentan las características del vector de análisis en el grupo de entrenamiento ejecutado por los nueve parámetros, donde el flujo espectral identificado en la figura como (Ft) presenta un sesgo correcto del espectro de potencia de las diferentes muestras y una media en los cuatro géneros de 0.1, obteniendo que los cuatro géneros no presentan mayor potencia en su muestras musicales, sin embargo el currulao presenta una mayor desviación en los datos debido a que es un genero en el que predomina la percusión de manera que se obtienen niveles mayores de potencia. Por otro lado, las características de atenuación (Ae) y detección de cruce por cero (Zt), presentan características similares para los cuatro géneros con una media en los datos para esas características de 0.2 y presentado una desviación de los datos cercanos a cinco, por lo cual se debe tener en cuenta la descendencia de los cuatro géneros donde su base es musical proviene de una combinación africana, española con una mezcla de los géneros nativos americanos. Por último se observa en el centroide que el joropo presenta el mayor brillo en la estructura de las piezas musicales, seguido por el vallenato y el currulao y dejando de último al bambuco debido a su composición secuencial en la instrumentación, teniendo una media de los datos para el centroide alrededor a uno y presentando una gran cambio en el brillo de los cuatro géneros.

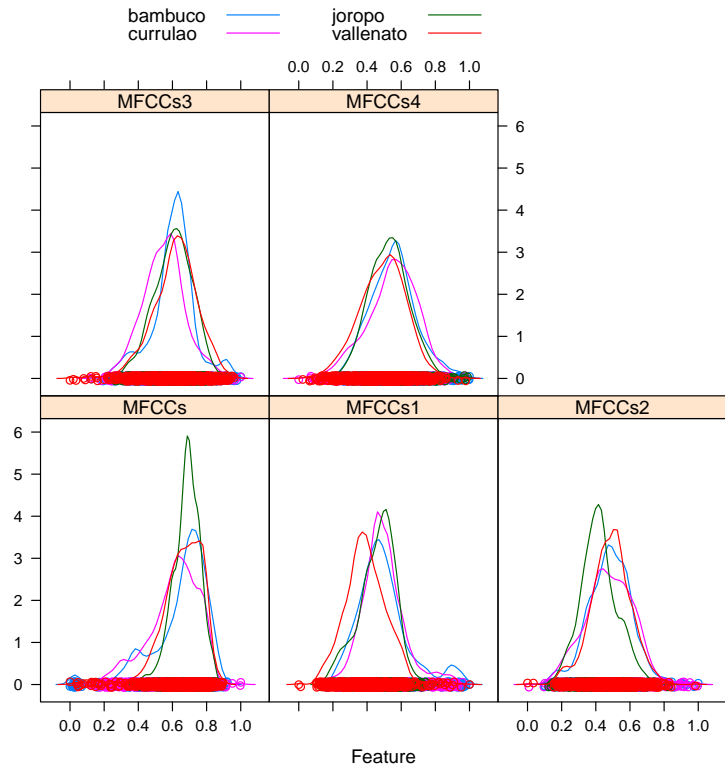


Figura 34: Densidad e identificación de características MFCC en el entrenamiento

De acuerdo al entrenamiento aplicado al algoritmo se obtiene la Figura 34 donde se presenta el realce de los coeficientes de MFCC, teniendo características que resaltan como el primer coeficiente MFCC, en donde se observa mayor cambio en la desviación de los datos y una media de los datos de 0.7, además teniendo en cuenta que es el coeficiente donde se presenta la energía con la cual el autor expresa su muestra musical, se obtiene una clara diferencia entre el joropo y los demás géneros, adicionalmente el bambuco siendo un genero donde los interpretes predominan sobre la instrumentación presenta un realce en la energía. Por otro lado, como se observa en los demás coeficientes MFCC se presenta una cierta similitud en cuanto a la desviación de los datos, indicando como se mencionó anteriormente la similitud en cuanto a las raíces de los géneros y presentado una media en los datos cercana a 0.6 de manera que los datos presentan cierta similitud en manejo de interpretación musical o interpretación por parte de la máquina en similitud de instrumentación.

Agregando a lo anterior y teniendo presente el análisis realizado en las secciones se ejecuta una elección del algoritmo clasificador donde se observan las características, el funcionamiento, el porcentaje de acierto en la clasificación de los cuatro géneros, la velocidad de ejecución y posterior mejoría del algoritmo para la clasificación de subgéneros folclóricos colombianos, donde se escoge el algoritmo soporte vector de maquina (SVM), el cual se presentan como un clasificador discriminatorio definido formalmente por un hiperplano de separación y el cual permite generar una optimización en la búsqueda de subgéneros folclóricos, por lo cual aun siendo el algoritmo de clasificación más lento presenta la mejor metodología y linealización de los cuatro géneros folclóricos.

Por otro lado, se observo que los algoritmos de aprendizaje presentan una relación para su óptimo funcionamiento siendo esto que los datos de entrada del sistema no deben ser mayores a los datos que ya se encuentran en el entrenamiento, adicionalmente los datos que ingresan como clasificación presentan una incertidumbre por lo cual los algoritmo no logran generar una clasificación si no se ingresa observaciones las cuales sea objeto de categorización, formado de esta manera un clasificador con ingreso de cuatro muestras musicales como grupo de clasificación para un total de 392 observaciones para su categorización.

8.4. Evaluar porcentaje precisión del algoritmo

8.4.1. Clasificación géneros folclóricos

El algoritmo de aprendizaje genera las siguientes matrices de confusión, de manera que se presenta como una comparación del grupo de entrenamiento respecto al grupo de muestras musicales que ingresa, por lo cual se realiza la predicción de los géneros y posteriormente se lleva a cabo el análisis de la matriz de confusión para los cuatro géneros a la vez como se observa en las siguientes figuras.

De acuerdo con la teoría de la sección 7.4.1, la matriz de confusión presenta datos como porcentaje de acierto, sensibilidad en la detección de los géneros, predicción positiva y negativa, donde se tiene presente la prevalencia de los datos que ingresan al sistema y la tasa de detección que presenta, posteriormente a la tasa de detección de la matriz se le realiza una regla simple de tres para determinar el porcentaje de acierto del algoritmo respecto a la muestra musical.

Como se puede observar en la Figura 36 y la Figura 37 se presenta un porcentaje de acierto para el grupo de clasificación el cual representa al valor medio para los cuatro géneros con una proporción de los datos del 33 % en el segundo grupo de clasificación y del 46 % para

Prediction	Reference			
	bambuco	currulao	joropo	vallenato
bambuco	27	22	2	1
currulao	8	42	9	0
joropo	26	2	57	2
vallenato	37	32	30	95

Overall Statistics

Accuracy : 0.5638
95% CI : (0.5131, 0.6135)
No Information Rate : 0.25
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.4184
McNemar's Test P-Value : < 2.2e-16

Statistics by Class:

	Class: bambuco	Class: currulao	Class: joropo
Sensitivity	0.27551	0.4286	0.5816
Specificity	0.91497	0.9422	0.8980
Pos Pred Value	0.51923	0.7119	0.6532
Neg Pred Value	0.79118	0.8318	0.8656
Prevalence	0.25000	0.2500	0.2500
Detection Rate	0.06888	0.1071	0.1454
Detection Prevalence	0.13265	0.1505	0.2219
Balanced Accuracy	0.59524	0.6854	0.7398

	Class: vallenato
Sensitivity	0.9694
Specificity	0.6633
Pos Pred Value	0.4897
Neg Pred Value	0.9848
Prevalence	0.2500
Detection Rate	0.2423
Detection Prevalence	0.4949
Balanced Accuracy	0.8163

Figura 35: Matriz de confusión algoritmo SVM.

Prediction	Reference			
	bambuco	currulao	joropo	vallenato
bambuco	59	2	22	38
currulao	18	14	36	32
joropo	12	36	35	4
vallenato	9	46	5	24

Overall Statistics

Accuracy : 0.3367
95% CI : (0.2901, 0.3859)
No Information Rate : 0.25
P-Value [Acc > NIR] : 7.513e-05

Kappa : 0.1156
McNemar's Test P-Value : 2.455e-06

Statistics by Class:

	Class: bambuco	Class: currulao	Class: joropo
Sensitivity	0.6020	0.14286	0.35714
Specificity	0.7891	0.70748	0.82313
Pos Pred Value	0.4876	0.14000	0.40230
Neg Pred Value	0.8561	0.71233	0.79344
Prevalence	0.25000	0.25000	0.25000
Detection Rate	0.1505	0.03571	0.08929
Detection Prevalence	0.3087	0.25510	0.22194
Balanced Accuracy	0.6956	0.42517	0.59014

	Class: vallenato
Sensitivity	0.24490
Specificity	0.79592
Pos Pred Value	0.28571
Neg Pred Value	0.75074
Prevalence	0.25000
Detection Rate	0.06122
Detection Prevalence	0.21429

Figura 36: Matriz de confusión algoritmo SVM.

el tercer grupo. Teniendo presente las características de la matriz de confusión se presenta la tabla 15, la tabla 16, la tabla 17 y la tabla 18; donde se especifica el porcentaje de acierto de cada canción, además es de tener en cuenta que se ingresan grupos de cuatro canciones en el orden de la tabla una por genero.

Ahora bien, dos parámetros como son la sensibilidad y especificidad permiten interpretar como el algoritmo determina la clasificación de las observaciones y los cuales indican la capacidad del algoritmo para discriminar los casos positivos de los negativos. Siendo la

```

      Reference
Prediction bambuco currulao joropo vallenato
bambuco      80      5      16      24
currulao      3      34      14      18
joropo       11      14      50      30
vallenato      4      45      18      26

Overall Statistics

      Accuracy : 0.4847
      95% CI : (0.4342, 0.5354)
      No Information Rate : 0.25
      P-Value [Acc > NIR] : < 2.2e-16

      Kappa : 0.3129
      McNemar's Test P-Value : 3.472e-05

Statistics by Class:

      Class: bambuco Class: currulao Class: joropo
Sensitivity      0.8163      0.34694      0.5182
Specificity      0.8469      0.88095      0.8129
Pos Pred Value   0.6400      0.49275      0.4762
Neg Pred Value   0.9326      0.80186      0.8328
Prevalence       0.2500      0.25000      0.2500
Detection Rate   0.2041      0.08673      0.1276
Detection Prevalence 0.3189      0.17602      0.2679
Balanced Accuracy 0.8316      0.61395      0.6616

      Class: vallenato
Sensitivity      0.26531
Specificity      0.77211
Pos Pred Value   0.27957
Neg Pred Value   0.75920
Prevalence       0.25000
Detection Rate   0.06633
Detection Prevalence 0.23724
Balanced Accuracy 0.51871

```

Figura 37: Matriz de confusión algoritmo SVM.

sensibilidad la que indica la relevancia de los resultados permitiendo obtener un algoritmo altamente exacto, mientras que un algoritmo específico determina la mayoría de resultados de interés. Teniendo esto presente la Figura 37 y la Figura 38, muestran que el algoritmo es altamente específico lo cual en la clasificación de muestras musicales permite tener un alto valor de acierto en la clasificación con una sensibilidad media permitiendo tener un algoritmo balanceado entre resultados relevante y resultados de interés.

```

      Reference
Prediction bambuco currulao joropo vallenato
bambuco      67      2      11      32
currulao     13      79      24      6
joropo       11      10      2      21
vallenato      7      7      61      39

Overall Statistics

      Accuracy : 0.477
      95% CI : (0.4267, 0.5278)
      No Information Rate : 0.25
      P-Value [Acc > NIR] : < 2.2e-16

      Kappa : 0.3027
      McNemar's Test P-Value : 6.07e-09

Statistics by Class:

      Class: bambuco Class: currulao Class: joropo
Sensitivity      0.6837      0.8061      0.020408
Specificity      0.8469      0.8537      0.857143
Pos Pred Value   0.5982      0.6475      0.045455
Neg Pred Value   0.8893      0.9296      0.724138
Prevalence       0.2500      0.2500      0.250000
Detection Rate   0.1709      0.2015      0.005102
Detection Prevalence 0.2857      0.3112      0.112245
Balanced Accuracy 0.7653      0.8299      0.438776

      Class: vallenato
Sensitivity      0.39796
Specificity      0.74490
Pos Pred Value   0.34211
Neg Pred Value   0.78777
Prevalence       0.25000
Detection Rate   0.09949
Detection Prevalence 0.29082
Balanced Accuracy 0.57143

```

Figura 38: Matriz de confusión algoritmo SVM.

Por tanto, se logró llegar a la parametrización de señales de audio en general, con los cuales algoritmos de aprendizaje automático ejecutan procesos de reconocimiento de patrones, permitiendo generar una herramienta que evite la clasificación errónea por parte del humano. Respecto a comparaciones hechas, precisamente con los antecedentes de esta misma caracterización en otros géneros, cabe resaltar el grano de arena que aporta este estudio al estado del arte de una investigación como ésta, poco vista y enfocada hacia el folclor colombiano.

8.4.2. Clasificación géneros populares

Con respecto al análisis realizado en el sección 8.4.1 se busca validar el entrenamiento del algoritmo con la clasificación de géneros populares como Rock, Pop, Balada, Salsa, Tropical, Merengue; realizando la comparación con una canción de los géneros folclóricos para completar un grupo de cuatro canciones como se observo en 7.3 de manera que se obtiene las siguientes matrices de confusión.

```

Accuracy : 0.1505
95% CI : (0.1166, 0.1898)
No Information Rate : 0.5
P-Value [Acc > NIR] : 1

Kappa : 0.0486
McNemar's Test P-Value : NA

Statistics by Class:

Class: bambuco Class: currulao Class: joropo
Sensitivity      0.6920      NA      NA
Specificity      0.6293      0.92602  0.824
Pos Pred Value   0.3512      NA      NA
Neg Pred Value   0.8259      NA      NA
Prevalence        0.2500      0.00000  0.000
Detection Rate    0.1505      0.00000  0.000
Detection Prevalence 0.4286      0.07398  0.176
Balanced Accuracy 0.6156      NA      NA

Class: vallenato Class: Balada Class: Merengue Class: Pop
Sensitivity      NA      0.0      NA      NA
Specificity      0.6786      1.0      1      1
Pos Pred Value   NA      NaN      NA      NA
Neg Pred Value   NA      0.5      NA      NA
Prevalence        0.0000      0.5      0      0
Detection Rate    0.0000      0.0      0      0
Detection Prevalence 0.3214      0.0      0      0
Balanced Accuracy NA      0.5      NA      NA

Class: Rock Class: Salsa Class: Tropical
Sensitivity      0.00      NA      NA
Specificity      1.00      1      1
Pos Pred Value   NaN      NA      NA
Neg Pred Value   0.75      NA      NA
Prevalence        0.25      0      0
Detection Rate    0.00      0      0
Detection Prevalence 0.00      0      0
Balanced Accuracy 0.50      NA      NA

```

Figura 39: Matriz de confusión algoritmo SVM.

En la Figura 39, se puede observar una prevalencia de los datos del 25 % y una tasa de detección del bambuco On tabas de Edmundo Arias con el 60,25 % respecto a las muestras musicales de balada, Lo mejor que hay de en mi vida y Magia de Andrés Cepeda estos con una prevalencia de los datos de ingreso del 50 % y tasa de detección de los géneros del 0 % y un rock en ingles Another one bites the dust con una prevalencia del 25 % y una tasa de detección del 0 %. Obteniendo que la media observada en la tabla 13 para la clasificación del bambuco es aproximada.

```

Accuracy : 0.2005
95% CI : (0.1621, 0.2435)
No Information Rate : 0.2487
P-Value [Acc > NIR] : 0.9898

Kappa : 0.1455
McNemar's Test P-Value : NA

Statistics by Class:

Class: bambuco Class: currulao Class: joropo
Sensitivity NA 0.8061 NA
Specificity 0.6853 0.9223 0.91371
Pos Pred Value NA 0.7745 NA
Neg Pred Value NA 0.9349 NA
Prevalence 0.0000 0.2487 0.00000
Detection Rate 0.0000 0.2005 0.00000
Detection Prevalence 0.3147 0.2589 0.08629
Balanced Accuracy NA 0.8642 NA

Class: vallenato Class: Balada Class: Merengue Class: Pop
Sensitivity NA 0.000000 NA 0.0000
Specificity 0.6599 1.000000 1 1.0000
Pos Pred Value NA NaN NA NaN
Neg Pred Value NA 0.994924 NA 0.7513
Prevalence 0.0000 0.005076 0 0.2487
Detection Rate 0.0000 0.000000 0 0.0000
Detection Prevalence 0.3401 0.000000 0 0.0000
Balanced Accuracy NA 0.500000 NA 0.5000

Class: Rock Class: Salsa Class: Tropical
Sensitivity 0.0000 0.0000 NA
Specificity 1.0000 1.0000 1
Pos Pred Value NaN NaN NA
Neg Pred Value 0.7513 0.7513 NA
Prevalence 0.2487 0.2487 0
Detection Rate 0.0000 0.0000 0
Detection Prevalence 0.0000 0.0000 0
Balanced Accuracy 0.5000 0.5000 NA

```

Figura 40: Matriz de confusión algoritmo SVM.

Como se puede observar en la Figura 40, se tiene el Currulao Adiós margarita de Canalon de timbiqui con una prevalencia de los datos de 25 % con una tasa de detección del 80,61 %, respecto a la muestra musical del Pop, The lazy song de Bruno Mars con una prevalencia de los datos del 25 % y una tasa de detección del 0 %, por otro lado se corren observaciones de la muestra musical La plata de Juanes como un rock y siendo una canción que el autor presenta como un vallenato el algoritmo lo detecta con una prevalencia de los datos del 25 % y una tasa de detección de 0 % esto de acuerdo al uso de instrumentación usado en la muestra musical y por ultimo una muestra musical del genero salsa, La rebelión de Alvaro José Arroyo con una prevalencia del 25 % y una tasa de detección del 0 %.

```

Accuracy : 0.1883
95% CI : (0.1508, 0.2305)
No Information Rate : 0.2519
P-Value [Acc > NIR] : 0.9989

Kappa : 0.1437
McNemar's Test P-Value : NA

Statistics by Class:

Class: bambuco Class: currulao Class: joropo
Sensitivity NA NA 0.7551
Specificity 0.6616 0.90585 0.9729
Pos Pred Value NA NA 0.9024
Neg Pred Value NA NA 0.9228
Prevalence 0.0000 0.00000 0.2494
Detection Rate 0.0000 0.00000 0.1883
Detection Prevalence 0.3384 0.09415 0.2087
Balanced Accuracy NA NA 0.8640

Class: vallenato Class: Balada Class: Merengue Class: Pop
Sensitivity NA NA NA 0.0000
Specificity 0.6412 1 1 1.0000
Pos Pred Value NA NA NA NaN
Neg Pred Value NA NA NA 0.7481
Prevalence 0.0000 0 0 0.2519
Detection Rate 0.0000 0 0 0.0000
Detection Prevalence 0.3588 0 0 0.0000
Balanced Accuracy NA NA NA 0.5000

Class: Rock Class: Salsa Class: Tropical
Sensitivity 0.0000 0.00000 0.0000
Specificity 1.0000 1.00000 1.0000
Pos Pred Value NaN NaN NaN
Neg Pred Value 0.7506 0.997455 0.7532
Prevalence 0.2494 0.002545 0.2468
Detection Rate 0.0000 0.00000 0.0000
Detection Prevalence 0.0000 0.00000 0.0000

```

Figura 41: Matriz de confusión algoritmo SVM.

A esto se añade la clasificación para el grupo donde se ingresa el joropo De que tamaño es tu amor de Reynaldo Armas, Figura 41, el cual presenta una prevalencia de los datos del 25 % y una tasa de detección del 75,32 %, respecto al grupo de géneros populares como el Pop, Thinking out loud de Ed Sheeran con una prevalencia en los datos 25 % y una tasa de detección del 0 %, el Rock mi verdad de Mana con una prevalencia del 25 % en los datos y una tasa de detección del 0 %; cerrando este grupo se encuentra una muestra musical tropical, Dame tu mujer José de los 50 de joselito con una prevalencia del 25 % y una tasa de detección del 0 %. observando que el algoritmo no obtiene información de clasificación en géneros comerciales y permitiendo ver la exactitud dada en la Tabla 13.

Accuracy : 0.2423				
95% CI : (0.2007, 0.2879)				
No Information Rate : 0.25				
P-Value [Acc > NIR] : 0.6554				
Kappa : 0.0973				
McNemar's Test P-Value : NA				
Statistics by Class:				
	Class: bambuco	Class: currulao	Class: joropo	
Sensitivity	NA	NA	NA	
Specificity	0.7372	0.96429	0.94133	
Pos Pred Value	NA	NA	NA	
Neg Pred Value	NA	NA	NA	
Prevalence	0.0000	0.00000	0.00000	
Detection Rate	0.0000	0.00000	0.00000	
Detection Prevalence	0.2628	0.03571	0.05867	
Balanced Accuracy	NA	NA	NA	
	Class: vallenato	Class: Balada	Class: Merengue	Class: Pop
Sensitivity	0.9694	0.00	0.00	0.00
Specificity	0.4660	1.00	1.00	1.00
Pos Pred Value	0.3770	NaN	NaN	NaN
Neg Pred Value	0.9786	0.75	0.75	0.75
Prevalence	0.2500	0.25	0.25	0.25
Detection Rate	0.2423	0.00	0.00	0.00
Detection Prevalence	0.6429	0.00	0.00	0.00
Balanced Accuracy	0.7177	0.50	0.50	0.50
	Class: Rock	Class: Salsa	Class: Tropical	
Sensitivity	NA	NA	NA	
Specificity	1	1	1	
Pos Pred Value	NA	NA	NA	
Neg Pred Value	NA	NA	NA	
Prevalence	0	0	0	
Detection Rate	0	0	0	
Detection Prevalence	0	0	0	
Balanced Accuracy	NA	NA	NA	

Figura 42: Matriz de confusión algoritmo SVM.

Por último la Figura 42, presenta la clasificación del grupo donde se categorizan el vallenato con respecto a los géneros populares, siendo, de niña a mujer de los chiches del vallenato la muestra musical con una prevalancia del 25 % de los datos de ingreso con una tasa de detección del 96,92 %, junto al genero folclórico se agrega la muestra musical del genero Pop, el arrepentido de Melendi y Carlos vives con una prevalencia de los datos del 25 % y una tasa de detección del 0 %, una balada, convénceme de Ricardo Montaner con una prevalencia del 25 % en el grupo de entrada y una tasa de detección de 0 % y como última muestra musical un merengue, No puedo olvidarla de ricarena con una prevalencia del 25 % y una tasa de detección del 0 %.

Como se puede observar en la sección 8.4.1, se obtiene la clasificación de 140 muestras musicales de los cuatro géneros con una media de categorización del 76,1224 % del bambuco, del 91,1564 % del currulao, del 83,7415 % del joropo y del 98 % del vallenato, siendo un porcentaje de acierto alto para la clasificación de los cuatro géneros folclóricos, por otro lado la clasificación obtenida en la categorización de géneros populares con respecto a los

géneros folclóricos se obtiene la Tabla 14

Titulo	Artista	Genero	Porcentaje acierto
On tabas	Alicia Edmundo Arias	Bambuco	60,20408 %
Lo mejor que hay de en mi vida	Andrés Cepeda	Balada	0 %
Magia	Andrés Cepeda	Balada	0 %
Another one bites the dust	Queen	Rock	0 %
Adios margarita	Canalón de timbiquí	Currulao	80,61 %
The lazy song	Bruno Mars	Pop	0 %
La plata	Juan Esteban Aristizabal	Rock	0 %
La rebelión	Alvaro José Arroyo	Salsa	0 %
De que tamaño es tu amor	Reynaldo Armas	Joropo	75,32 %
Thinking out loud	Ed Sheeran	Pop	0 %
Mi verdad	Mana & Shakira	Rock	0 %
Dame tu mujer José	50 Jodelito	Tropical	0 %
De niña a mujer	Los chiches del vallenato	Vallenato	96,92 %
El arrepentido	Melendi & Carlos Vives	Pop	0 %
Convénceme	Ricardo Montaner	Balada	0 %
No puedo olvidarla	Ricarena	Merengue	0 %

Tabla 14: Nombres muestras musical clasificación géneros populares

Por tanto se obtiene un algoritmo clasificador de cuatro géneros folclóricos con un porcentaje de acierto alto, generando de esta manera una herramienta de clasificación automática mediante audición por computadora, la cual no logra distinguir entre subgéneros que se presentan en la clasificación y entrenamiento del algoritmo, como son el San juanero, el currulao bambuco, el pasaje, el son, la puya, el paseo o el merengue; siento esto un proceso de mejora en posteriores clasificadores.

9. CONCLUSIONES Y TRABAJOS FUTUROS

9.1. Conclusiones

Este trabajo se diseñó una herramienta de audición computacional para la identificación de música folclórica colombiana y la separación de cuatro géneros musicales, Bambuco, Currulao, Joropo y Vallenato, basada en una de las tres dimensiones musicales, la textura tímbrica. Esta dimensión musical se evalúa a través de características de bajo y medio nivel extraídas de la señal de audio. Sus valores se utilizan en un algoritmo de vector soporte de máquina con reglas explícitas como el kernel radial, para obtener una clasificación múltiple de extracto de música. El sistema desarrollado puede entonces verse como compuesto de dos bloques principales: un extractor de características y un clasificador múltiple donde el extractor de características alimenta información al clasificador.

Para el sistema implementado, se eligieron los valores de Centroide espectral y Flujo espectral como características que resaltan y se usan para evaluar esta dimensión. Dado que las características de nivel medio están más relacionadas con los aspectos musicales de la señal, desarrollado por (Muñoz, 2016), de manera que presentan las características de bajo nivel como aquellas que están relacionadas con los aspectos del timbre de la señal que se reflejan, por ejemplo, en la instrumentación y la voz, siendo esto las características con las cuales se realiza una observación sobre las cuales se desarrollan los tres métodos de aprendizaje donde la estructura rítmica de la señal se analiza en función de la auto correlación que detecta las periodicidades de la señal en función de los retrasos donde la auto correlación muestra picos, estos retrasos corresponden a una identificación de los instantes de tiempo donde la señal es más similar a sí misma y de los cuales se pueden hacer uso para una optimización en la clasificación de los géneros o posibles subgéneros.

Las características usadas para la clasificación podrían funcionar para medidas de similitud, siendo usadas por cualquier técnica de clasificación no supervisada. Por ejemplo, se podría encontrar canciones con la distancia euclidiana más pequeña a la canción de entrenamiento en el espacio definido por las funciones proyectadas en sus componentes principales. El éxito del clasificador k -NN indica que la mayoría de estas canciones son realmente similares, de manera que se puede confiar en las etiquetas de género como una medida de similitud.

Algunas de las características utilizadas en la sección 7.1 se presentan algunas como redundantes y podrían descartarse como algunos de los coeficientes de Mel sin disminuir la precisión de la clasificación, esto ahorraría tiempo y memoria durante la extracción, el

entrenamiento y la clasificación de las características, y reduciría los requisitos de almacenamiento para los algoritmos simples como k -NN. Donde la dimensionalidad reducida también podría aumentar la precisión de la clasificación en algoritmos como los vectores soporte de maquina.

Con este trabajo se clasificó y caracterizó la música folclórica colombiana mediante los parámetros ya estipulados y que han sido aplicados a diferentes géneros del mundo, donde se han encontrado diferentes factores, los cuales dejan todavía una gran incertidumbre en el entendimiento y universalidad del folclor musical, de manera que se busca optimizar el porcentaje de acierto del algoritmo; tema discutido en el capítulo sobre los inconvenientes en la clasificación musical, donde se establece que al realizar dicha clasificación no es tener en cuenta únicamente el lenguaje musical, también los diferentes factores sociales, culturales, históricos, hasta políticos y económicos, causantes de confusión como la diferencia entre los términos género y ritmo, incluso a través de la historia los mismos autores y la fusión sonora que los medios masivos de comunicación y hasta la misma gente se encarga de hacer, no sólo en la música colombiana si no en general.

El algoritmo SVM se caracteriza en gran medida por la elección del kernel donde el SVM vinculan los problemas para los que están diseñados con una gran cantidad de trabajo existente sobre métodos basados en el kernel proporcionado un nuevo enfoque al problema del reconocimiento de patrones con conexiones claras a la teoría del aprendizaje estadístico subyacente. De manera que quizás la mayor limitación del enfoque del vector de soporte reside en la elección del núcleo. Una vez que se arregla el kernel, los clasificadores SVM solo tienen un parámetro elegido por el usuario siendo el costo de clasificación y en donde se presenta una limitación en la velocidad y el tamaño, tanto en el entrenamiento como en las pruebas.

En este proceso se busca obtener las muestras musicales con la mejor calidad posible teniendo presente que varias de las grabaciones son de disco LP y en la digitalización presentan un gran cantidad de ruido, de manera que se descartan estas pistas para de esta manera obtener los mejores resultados. Adicional se tiene presente que géneros como el Bambuco y el Currulao, cuentan con representantes modernos donde las grabaciones no son de calidad debido a que son estudiantinas y autores emergentes, por otro lado, los géneros Joropo y Vallenato tiene muestras de audio de mayor calidad lo cual se observa en los resultados de la clasificación.

Como comentario general, se puede decir que el sistema desarrollado presenta una buena eficiencia. Se basa en un nueve características en el entrenamiento del algoritmo SVM, lo que lleva a un tiempo de cálculo reducido y donde los resultados de precisión son muy

buenos, con 81 % de precisión para la base de datos compilada, teniendo en cuenta la simplicidad del sistema. La mejora de la precisión debe considerarse como un trabajo futuro, pero solo será posible a costa de sacrificar la simplicidad, tiempo de cálculo.

9.2. Trabajos futuros

El nivel de rendimiento de este sistema puede servir como base para futuros trabajos en la clasificación de los cerca de 100 géneros y subgéneros folclóricos colombianos, de manera que el software está disponible para los investigadores y desarrolladores que deseen realizar más pruebas o desarrollarlo. Ahondando en las dimensiones de contenido armónico o contenido rítmico, se puede obtener un clasificador con un grado de acierto mayor, esto permitido mediante características que no se tuvieron en cuenta en este proyecto.

Los algoritmos más precisos y robustos provendrán de enfoques más firmemente arraigados en la psicoacústica y el análisis de la escena auditiva, donde se tienen en cuenta estado de ánimo, selectividad en la escritura de la letra entre otras características que se tienen en la psicoacústica; pero hasta que se desarrollen tales sistemas, los algoritmos más simples como este pueden ser de valor práctico.

Por último para mejorar aún más el modelo, se puede probar la misma serie de pasos de optimización con diferentes algoritmos de aprendizaje, ya que la mejora de los distintos resultados de las técnicas de optimización varía con el algoritmo. Puede repetir el proceso iterativo descrito en la sección 8, por ejemplo, revisando el algoritmo k -NN, que funcionó bien inicialmente. Incluso volviendo a la fase de extracción de características y buscar características adicionales. Para identificar el mejor modelo, siempre es necesario iterar entre las diferentes fases del flujo de trabajo de aprendizaje automático.

Referencias

- Azure, M. (2019). *Cómo elegir algoritmos*. Descargado de <https://docs.microsoft.com/es-es/azure/machine-learning/studio/algorithm-choice>
- Brownlee, J. (2016). *Naive bayes for machine learning*. Descargado de <https://machinelearningmastery.com/naive-bayes-for-machine-learning/>
- Castro Rey, C. (2018). Clasificación y su implementación en r.
- Center, I. K. (2018). *Funcionamiento de svm*. Descargado de https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/clementine/svm_howwork.html
- Chowning, J., Rush, L., Mont-Reynaud, B., Chafe, C., Schloss, W., y Smith, J. (1984, jan). *Intelligent systems for the analysis of digitized acoustic signal*. Center for Computer Research in music and acoustic.
- Co colombia. (2017, apr). Descargado de <http://www.colombia.travel/es>
- Coursera. (2018). *Audio signal processing for music applications*. Descargado de <https://es.coursera.org/lecture/audio-signal-processing/stft-N8Trk>
- dos Reis, G. M. J. (2014). *Una aproximación genética a la transcripción automática de música* (Tesis Doctoral no publicada). Universidad de Extremadura.
- Educational, U. N. (2015). *El vallenato, música tradicional de la región del magdalena grande*. Descargado de <https://ich.unesco.org/es/USL/el-vallenato-musica-tradicional-de-la-region-del-magdalena-grande-01095>
- Foote, J., y Uchihashi, S. (2001, 08). The beat spectrum: A new approach to rhythm analysis. En (Vol. 22). doi: 10.1109/ICME.2001.1237863
- García, E., Pacheco, J., y Mancera, A. (2005). Clasificación de música por género utilizando redes neuronales artificiales. En *Ciic* (pp. 284–291).
- García, E., Pacheco, G., y Mancera, G. (2005). Clasificación de música por género utilizando redes neuronales artificiales. *Memorias del Congreso Internacional de Inteligencia Computacional*, 1–9.
- Giraldo-Guzmán, J., Marrugo, A. G., y Contreras-Ortiz, S. H. (2016). Vehicle speed estimation using audio features and neural networks. En *Andescon, 2016 ieee* (pp. 1–4).
- González, L. (2018). *Aprendizaje supervisado: K-nearest neighbors*. Descargado de <http://ligdigonzalez.com/aprendizaje-supervisado-k-nearest-neighbors/>
- Gouyon, F. (2006). *A computational approach to rhythm description-audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing*. Universitat Pompeu Fabra.

- Igor, K., y Subramanian, D. (2002). Hidden markov classification for musical genres. *Course Project*.
- Karatzoglou, A., Meyer, D., y Hornik, K. (2005). Support vector machines in R.
- Khan, R. (2017). *Naive bayes classifier: theory and r example*. Descargado de https://rpubs.com/riazakhan94/naive_bayes_classifier_e1071
- Lantz, B. (2013). *Machine learning with r*. Packt Publishing Ltd.
- López, A. C., y Molina, L. P. (2017). *Clasificador de subgéneros de música electrónica*. Descargado de <https://eprints.ucm.es/44672/> (Trabajo de Fin de Grado en Ingeniería Informática (Universidad Complutense, Facultad de Informática, curso 2016/2017))
- López García, M. V. (2014). *Análisis de estructura temporal de datos musicales para clasificación* (B.S. thesis).
- Mathworks. (2018). *Machine learning tres cosas que es necesario saber*. Descargado de <https://la.mathworks.com/discovery/machine-learning.html>
- Mejía Gallego, M. V. (2012). *Una gran aventura por colombia: planteamiento conceptual de un material multimedia interactivo y didáctico de cinco ritmos tradicionales colombianos para niños de 4 a 7 años de edad [recurso electrónico]* (Tesis Doctoral no publicada).
- Melo Ángel, C. (s.f.). *El documento musical en los procesos de salvaguardia del patrimonio musical colombiano el centro de documentación musical del ministerio de cultura* (Tesis de Master no publicada). Facultad de Arquitectura y Diseño.
- Muñoz, S. G. (2016). *Algoritmo para la caracterización por género musical basado en análisis y tratamiento de señales de audio en géneros musicales folclóricos colombianos* (software).
- Paradzinets, A., Harb, H., y Chen, L. (2009). Multiexpert system for automatic music genre classification. *Teknik Rapor, Ecole Centrale de Lyon, Departement MathInfo*.
- Pedraza, C., Vitola, J., Sepulveda, J., y Martinez, J. I. (2013). Fast content-based audio retrieval algorithm. En *Image, signal processing, and artificial vision (stiva), 2013 xviii symposium of* (pp. 1–5).
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Quevedo Urrea, J. H. (2011). El documento musical en colombia: la develación de una memoria oculta que la cartografía de prácticas musicales en colombia ha convertido en una estrategia de investigación y conocimiento. *Signo y Pensamiento*, 31(59).
- Rodrigo, J. A. (2017). *Clustering y heatmaps: aprendizaje no supervisado*. Descargado de https://rpubs.com/Joaquin_AR/310338

- Rodríguez-Quiles, J. A., y cols. (2011). Música, participación, identidad: nuevos retos en la era digital. *Revista interuniversitaria de formación del profesorado*(71), 175.
- Sirohi, K. (2018). *K-nearest neighbors algorithm with examples in r (simply explained knn)*. Descargado de <https://towardsdatascience.com/k-nearest-neighbors-algorithm-with-examples-in-r-simply-explained-knn-1f2c88da405c>
- Spotify. (2018). *Playlists*. Descargado de <https://support.spotify.com/cr/using-spotify/playlists/create-a-playlist/>
- Tzanetakis, G., Essl, G., y Cook, P. (2002). Automatic musical genre classification. *IEEE Transactions on Speech and Audio Processing*, 293–302.
- Uribe, A. V. (2002). *Hacia un estado comunitario (2002-2006)*. Descargado de <https://www.dnp.gov.co/Plan-Nacional-de-Desarrollo/Paginas/Planes-de-Desarrollo-anteriores.aspx>
- Wikipedia. (2018a). *Navaja de ockham — wikipedia, la enciclopedia libre*. Descargado de https://es.wikipedia.org/w/index.php?title=Navaja_de_Ockham&oldid=116187586 ([Internet; descargado 11-noviembre-2018])
- Wikipedia. (2018b). *Teorema de muestreo de nyquist-shannon — wikipedia, la enciclopedia libre*. Descargado de https://es.wikipedia.org/w/index.php?title=Teorema_de_muestreo_de_Nyquist-Shannon&oldid=110240822 ([Internet; descargado 4-julio-2019])
- Zelada, C. (2017). *Evaluación de modelos de clasificación*. Descargado de <https://rpubs.com/chzelada/275494>

ANEXO

A. Configuración youtube-dl

youtube-dl es un programa de línea de comandos para descargar videos de YouTube.com y algunos sitios más. Requiere el intérprete de Python, versión 2.6, 2.7 o 3.2+, y no es específico de la plataforma. Debería funcionar en Unix, en Windows o en macOS. Es de dominio público.

```
1 youtube-dl [OPTIONS] URL [URL...]
```

Listing 12: Ejemplos comandos youtube-dl

Puede configurar youtube-dl colocando cualquier opción de línea de comandos compatible en un archivo de configuración. En Linux y macOS, el archivo de configuración de todo el sistema se encuentra en `/etc/youtube-dl.conf` y el archivo de configuración de todo el usuario en `./config / youtube-dl / config`. En Windows, las ubicaciones de los archivos de configuración de todo el usuario son `% APPDATA% youtube-dl config.txt` o `C: Users <nombre de usuario> youtube-dl.conf`. Tenga en cuenta que, de forma predeterminada, es posible que el archivo de configuración no exista, por lo que es posible que deba crearlo usted mismo.

Por ejemplo, con el siguiente archivo de configuración, youtube-dl siempre extraerá el audio, no copiará el mtime, usará un proxy y guardará todos los videos en el directorio de películas en su directorio de inicio:

```
1 # Lines starting with # are comments
2
3 # Always extract audio
4 -x
5
6 # Do not copy the mtime
7 --no-mtime
8
9 # Use this proxy
10 --proxy 127.0.0.1:3128
11
12 # Save all videos under Movies directory in your home directory
13 -o ~/Movies/%(title)s.%(ext)s
```

Listing 13: Comandos configuración youtube-dl

Tenga en cuenta que las opciones en el archivo de configuración son simplemente las mismas opciones que los interruptores que se usan en las llamadas de la línea de comandos

regulares, por lo que no debe haber espacios en blanco después - or --, e.g. -o or --proxy pero no - o or -- proxy.

Puede usar `--ignore-config` si desea deshabilitar el archivo de configuración para una ejecución particular de youtube-dl.

También puede usar `--config-location` si desea usar un archivo de configuración personalizado para una ejecución particular de youtube-dl.

Instalación Para instalarlo para todos los usuarios de UNIX (Linux, macOS, etc.), escriba:

```
1 sudo curl -L https://yt-dl.org/downloads/latest/youtube-dl -o /usr/local/bin/youtube-dl
2 sudo chmod a+rx /usr/local/bin/youtube-dl
```

Listing 14: Comandos instalación curl youtubedl

Si no tienes curl, puedes usar alternativamente un wget reciente:

```
1 sudo wget https://yt-dl.org/downloads/latest/youtube-dl -O /usr/local/bin/youtube-dl
2 sudo chmod a+rx /usr/local/bin/youtube-dl
```

Listing 15: Comandos instalación wget youtubedl

Salida La opción `-o` permite a los usuarios indicar una plantilla para los nombres de archivo de salida.

El uso básico no es establecer ningún argumento de plantilla al descargar un solo archivo, como en `youtube-dl -o funny_video.flv 'https://some/video'`. Sin embargo, puede contener secuencias especiales que se reemplazarán al descargar cada video. Las secuencias especiales se pueden formatear de acuerdo con las operaciones de formateo de cadenas de python. Por ejemplo, `% (NAME)s` ó `% (NAME)05d`. Para aclarar, es un símbolo de porcentaje seguido de un nombre entre paréntesis, seguido de operaciones de formato. Los nombres permitidos junto con el tipo de secuencia son:

- `id (string)`: Identificador de vídeo
- `title (string)`: Título del vídeo
- `url (string)`: URL del vídeo

- `ext` (string): Extensión de archivo de vídeo
- `alt_title` (string): Un título secundario del vídeo
- `display_id` (string): Un identificador alternativo para el vídeo
- `uploader` (string): Nombre completo del cargador de vídeo
- `license` (string): Nombre de la licencia bajo la cual se licencia el vídeo
- `creator` (string): El creador del video.
- `release_date` (string): La fecha (AAAAMMDD) en que se publicó el vídeo
- `timestamp` (numeric): UNIX timestamp del momento en que el vídeo estuvo disponible
- `upload_date` (string): Fecha de carga del vídeo (AAAAMMDD)
- `uploader_id` (string): Apodo o id del cargador de vídeo
- `channel` (string): Nombre completo del canal en el que se carga el video
- `channelId` (string): Id del canal
- `location` (string): Ubicación física donde se filmó el vídeo
- `duration` (numeric): Duración del vídeo en segundos.
- `view_count` (numeric): Cuántos usuarios han visto el vídeo en la plataforma
- `like_count` (numeric): Número de valoraciones positivas del vídeo
- `dislike_count` (numeric): Número de calificaciones negativas del vídeo
- `repost_count` (numeric): Número de repeticiones del vídeo
- `average_rating` (numeric): Calificación promedio otorgada por los usuarios, la escala utilizada depende de la página web
- `comment_count` (numeric): Número de comentarios en el vídeo
- `age_limit` (numeric): Restricción de edad para el vídeo (años)
- `is_live` (boolean): Si este vídeo es una transmisión en vivo o un vídeo de duración fija
- `start_time` (numeric): Tiempo en segundos en que debe comenzar la reproducción, como se especifica en la URL

- `end_time` (numeric): Tiempo en segundos en que debe finalizar la reproducción, como se especifica en la URL
- `format` (string): Una descripción legible por humanos del formato
- `format_id` (string): Código de formato especificado por `--format`
- `format_note` (string): Información adicional sobre el formato
- `width` (numeric): Ancho del vídeo
- `height` (numeric): Altura del vídeo
- `resolution` (string): Descripción textual de ancho y alto
- `tbr` (numeric): Bitrate promedio de audio y vídeo en KBit/s
- `abr` (numeric): Velocidad de bits de audio promedio en KBit/s
- `acodec` (string): Nombre del códec de audio en uso
- `asr` (numeric): Frecuencia de muestreo de audio en hercios
- `vbr` (numeric): Velocidad de bits de vídeo promedio en KBit/s
- `fps` (numeric): Velocidad de fotogramas
- `vcodec` (string): Nombre del códec de vídeo en uso
- `container` (string): Nombre del formato del contenedor
- `filesize` (numeric): El número de bytes, si se conoce de antemano
- `filesize_approx` (numeric): Una estimación para el número de bytes
- `protocol` (string): El protocolo que se utilizará para la descarga real
- `extractor` (string): Nombre del extractor
- `extractor_key` (string): Nombre de clave del extractor
- `epoch` (numeric): Unix epoch al crear el archivo
- `autonumber` (numeric): Número de cinco dígitos que se incrementará con cada descarga, comenzando desde cero
- `playlist` (string): Nombre o id de la lista de reproducción que contiene el vídeo

- `playlist_index` (numeric): Índice del vídeo en la lista de reproducción rellenado con ceros iniciales según la longitud total de la lista de reproducción
- `playlist_id` (string): Identificador de lista de reproducción
- `playlist_title` (string): Título de playlist
- `playlist_uploader` (string): Nombre completo del cargador de la lista de reproducción
- `playlist_uploader_id` (string): Apodo o id del cargador de la lista de reproducción

Cada secuencia mencionada hace referencia a una plantilla de salida la cual será reemplazada por el valor real correspondiente al nombre de la secuencia. Teniendo en cuenta que no se garantiza que algunas de las secuencias estén presentes, ya que dependen de los metadatos obtenidos por un extractor en particular. Tales secuencias serán reemplazadas con NA.

Por ejemplo, para `-o%(title)s-%(id)s.%(ext)s` y un video mp4 con título youtube-dl test vídeo e id BaW_jenozKcj esto resultará en un vídeo de prueba `youtube-dl-BaW_jenozKcj`. Archivo mp4 creado en el directorio actual.

Las plantillas de salida también pueden contener una ruta jerárquica arbitraria, por ejemplo, `-o '%(playlist)s/%(playlist_index)s - %(title)s.%(ext)s'`, que resultará en la descarga de cada vídeo en un directorio correspondiente a esta plantilla de ruta. Cualquier directorio faltante se creará automáticamente para usted.

Ejemplos de plantillas de salida Tenga en cuenta que en Windows necesita usar comillas dobles en lugar de simples.

```

1
2 $ youtube-dl --get-filename -o '%(title)s.%(ext)s' BaW\_jenozKc --restrict-
   filenames
3 youtube-dl_test_video_.mp4           # A simple file name
4
5 # Download YouTube playlist videos in separate directory indexed by video
   order in a playlist
6 $ youtube-dl -o '%(playlist)s/%(playlist_index)s - %(title)s.%(ext)s' https
   ://www.youtube.com/playlist?list=PLwiyx1dc3P2JR9N8gQaQN_BCv1Slap7re
7
8 # Download all playlists of YouTube channel/user keeping each playlist in
   separate directory:
9 $ youtube-dl -o '%(uploader)s/%(playlist)s/%(playlist_index)s - %(title)s
   .%(ext)s' https://www.youtube.com/user/TheLinuxFoundation/playlists

```

```

10
11 # Download Udemy course keeping each chapter in separate directory under
    MyVideos directory in your home
12 $ youtube-dl -u user -p password -o '~/MyVideos/%(playlist)s/%(
    chapter_number)s - %(chapter)s/%(title)s.%(ext)s' https://www.udemy.com
    /java-tutorial/
13
14 # Download entire series season keeping each series and each season in
    separate directory under C:/MyVideos
15 $ youtube-dl -o "C:/MyVideos/%(series)s/%(season_number)s - %(season)s/%(
    episode_number)s - %(episode)s.%(ext)s" https://videomore.ru/
    kino_v_detalayah/5_sezon/367617
16
17 # Stream the video being downloaded to stdout
18 $ youtube-dl -o - BaW_jenozKc

```

Listing 16: Ejemplos plantilla de salida

Selección formato: Por defecto, youtube-dl intenta descargar la mejor calidad disponible, es decir, si desea obtener la mejor calidad que no necesita para aprobar ninguna opción especial, youtube-dl lo adivinará por defecto.

Pero a veces es posible que desee descargar en un formato diferente, por ejemplo, cuando está en una conexión lenta o intermitente. El mecanismo clave para lograr esto es la llamada selección de formato en función de la cual puede especificar explícitamente el formato deseado, seleccionar formatos en función de algún criterio o criterios, la prioridad de configuración y mucho más.

La sintaxis general para la selección de formato es `--format FORMAT` o más corto `-f FORMATO` donde `FORMATO` es una expresión de selección, es decir, una expresión que describe el formato o los formatos que desea descargar.

El caso más simple es solicitar un formato específico, por ejemplo, con `-f22` puede descargar el formato con un código de formato igual a 22. Puede obtener la lista de códigos de formato disponibles para un video en particular usando `--list-formats` ó `-F`. Tenga en cuenta que estos códigos de formato son específicos del extractor. `-f webm` descargará el mejor formato de calidad con la extensión webm como un solo archivo.

También puede utilizar nombres especiales para seleccionar formatos de casos de borde particulares:

- mejor: seleccione el mejor formato de calidad representado por un solo archivo con

vídeo y audio.

- Peor: seleccione el formato de peor calidad representado por un solo archivo con vídeo y audio.
- bestvideo: seleccione el formato de mejor calidad de vídeo solo (por ejemplo, vídeo DASH). Puede que no esté disponible.
- worstvideo: seleccione el formato de vídeo de peor calidad. Puede que no esté disponible.
- bestaudio: Seleccione el formato de audio de mejor calidad. Puede que no esté disponible.
- worstaudio: seleccione el formato de audio de peor calidad. Puede que no esté disponible.

Por ejemplo, para descargar el formato de vídeo de peor calidad, puede usar `-f worstvideo`.

Si desea descargar varios vídeos y no tienen los mismos formatos disponibles, puede especificar el orden de preferencia con barras diagonales. Tenga en cuenta que la barra inclinada es asociativa a la izquierda, es decir, se prefieren los formatos en el lado izquierdo, por ejemplo. `-f 22/17/18` descargará el formato 22 si está disponible, de lo contrario descargará el formato 17 si está disponible, de lo contrario descargará el formato 18 si está disponible, de lo contrario se quejará de que no hay formatos adecuados disponibles para descargar.

Si desea descargar varios formatos del mismo vídeo, use una coma como separador, por ejemplo. `-f 22,17,18` descargará estos tres formatos, por supuesto, si están disponibles. O un ejemplo más sofisticado combinado con la función de precedencia: `-f 136/137/mp4/bestvideo,140`

B. Algoritmo complemento extracción características

B.1. Octave

```
1
2 function cell2csv(fileName, cellArray, separator, excelYear, decimal)
3 % Writes cell array content into a *.csv file.
4 %
5 % CELL2CSV(fileName, cellArray, separator, excelYear, decimal)
6 %
7 % fileName      = Name of the file to save. [ i.e. 'text.csv' ]
8 % cellArray     = Name of the Cell Array where the data is in
9 % separator     = sign separating the values (default = ';')
10 % excelYear     = depending on the Excel version, the cells are put into
11 %                quotes before they are written to the file. The separator
12 %                is set to semicolon (;)
13 % decimal       = defines the decimal separator (default = '.')
14 %
15 %                by Sylvain Fiedler, KA, 2004
16 % updated by Sylvain Fiedler, Metz, 06
17 % fixed the logical-bug, Kaiserslautern, 06/2008, S.Fiedler
18 % added the choice of decimal separator, 11/2010, S.Fiedler
19
20 %% Checking for optional Variables
21 if ~exist('separator', 'var')
22     separator = ',';
23 end
24
25 if ~exist('excelYear', 'var')
26     excelYear = 1997;
27 end
28
29 if ~exist('decimal', 'var')
30     decimal = '.';
31 end
32
33 %% Setting separator for newer excelYears
34 if excelYear > 2000
35     separator = ';';
36 end
37
38 %% Write file
39 datei = fopen(fileName, 'w');
40
41 for z=1:size(cellArray, 1)
42     for s=1:size(cellArray, 2)
```

```

43
44     var = eval(['cellArray{z,s}']);
45     % If zero, then empty cell
46     if size(var, 1) == 0
47         var = '';
48     end
49     % If numeric -> String
50     if isnumeric(var)
51         var = num2str(var);
52         % Conversion of decimal separator (4 Europe & South America)
53         % http://commons.wikimedia.org/wiki/File:DecimalSeparator.svg
54         if decimal ~= '.'
55             var = strrep(var, '.', decimal);
56         end
57     end
58     % If logical -> 'true' or 'false'
59     if islogical(var)
60         if var == 1
61             var = 'TRUE';
62         else
63             var = 'FALSE';
64         end
65     end
66     % If newer version of Excel -> Quotes 4 Strings
67     if excelYear > 2000
68         var = ['"' var '"'];
69     end
70
71     % OUTPUT value
72     fprintf(datei, '%s', var);
73
74     % OUTPUT separator
75     if s ~= size(cellArray, 2)
76         fprintf(datei, separator);
77     end
78 end
79 if z ~= size(cellArray, 1) % prevent a empty line at EOF
80     % OUTPUT newline
81     fprintf(datei, '\n');
82 end
83 end
84 % Closing file
85 fclose(datei);
86 % END

```

Listing 17: Cell2csv

```

1 function cell2file( list, file )
2 % CELL2FILE Write cell array of strings to file.
3 %
4 %   CELL2FILE(LIST,FILE) writes each element of a cell array of strings
5 %   (columnwise) as lines to an ASCII file.
6 %
7 %   Example
8 %       % cell array of strings
9 %       list = { 'audio1.wav', 'audio2.wav', 'audio3.wav' };
10 %
11 %       % output filename
12 %       file = 'file.txt';
13 %
14 %       % write each line of list to file
15 %       cell2file( list, file );
16 %
17 %       % verify contents of the file
18 %       if isunix, system(sprintf('cat %s',file));
19 %       else, edit(file); end;
20 %
21 %   See also FILE2CELL.
22
23 %   Author: Kamil Wojcicki, June 2011
24
25
26 % very lite input validation
27 if nargin~=2, error(sprintf('See usage information:\n help %s',
mfilename)); end;
28
29 % open an ASCII file for writing, overwrite if exists
30 fid = fopen( file, 'w+' );
31
32 % write each element of list (columnwise) to file
33 fprintf( fid, '%s\n', list{:} );
34
35 % clean up
36 fclose( fid );
37
38
39 % EOF

```

Listing 18: cell2file

```

1 % Copyright (C) Ingenieria Electronica. 2018 UCundinamarca
2 %
3 % Calcula la Atenuacion espectral de @var{vector}
4 %

```

```

5 % Author: Sebastian Munoz <sebastianmg.info@gmail.com>
6 % Created: 2016
7
8 % modified by: Jose Zamora <learningtesis@hotmail.com>
9 % Date: 2018
10
11 function[Ae]=atenuacion(Tw,Ts,fs,vector)
12
13 Nw=round(1E-3*Tw*fs); % duracion del marco en muestras
14 Ns=round(1E-3*Ts*fs); % duracion del cambio de marco en muestras
15
16 nfft=2^nextpow2(Nw); % Longitud del analisis FFT
17
18 % Enmarcado y ventanamiento (marcos como columnas)
19 frames=vec2frames(vector,Nw,Ns);
20
21 % Calculo del espectro de magnitud (como vectores columna)
22 MAG=abs(fft(frames,nfft,1));
23
24 Ae=0.85.*(sum(MAG)); % Atenuacion Espectral
25
26 end

```

Listing 19: Atenuacion espectral

```

1 % Copyright (C) Ingenieria Electronica. 2018 UCundinamarca
2 %
3 % Calcula el centroide espectral de @var{vector}
4 %
5 % Author: Sebastian Munoz <sebastianmg.info@gmail.com>
6 % Created: 2016
7
8 % modified by: Jose Zamora <learningtesis@hotmail.com>
9 % Date: 2018
10 function[Ct]=centroide(Tw,Ts,fs,vector)
11
12 %% verificar parametros
13 Nw=round(1E-3*Tw*fs); % duracion del marco en muestras
14 Ns=round(1E-3*Ts*fs); % duracion del cambio de marco en muestras
15 nfft=2^nextpow2(Nw); % Longitud del analisis FFT
16
17 % Enmarcado y ventanamiento (marcos como columnas)
18 frames=vec2frames(vector,Nw,Ns);
19 % Calculo del espectro de magnitud (como vectores columna)
20 MAG=abs(fft(frames,nfft,1));
21
22 m=size(MAG);

```

```

23 t=1:m(2);
24 n=1:m(1);
25 n=n';
26 mag_n=(MAG).*(n);
27
28 num=sum(mag_n);
29 den=sum(MAG);
30 Ct=num./den; % Centroide Espectral
31
32 end

```

Listing 20: Centroide

```

1 % ## Copyright (C) Ingenieria Electronica. 2018 Ucundinamarca
2
3 % Deteccion del cruce por cero
4
5 % ## Author: Sebastian Munoz <sebasmg.reggae@gmail.com>
6 % ## Created: 2016
7 % ## Modified by: Jose Zamora <learningtesis@hotmail.com>
8 % ## Date: 2018
9 function [Zt_sum] = cruce(Tw, Ts, fs, vector)
10
11 Nw = round( 1E-3*Tw*fs ); % frame duration (samples)
12 Ns = round( 1E-3*Ts*fs ); % frame shift (samples)
13
14 nfft = 2^nextpow2( Nw ); % length of FFT analysis
15
16 % Framing and windowing (frames as columns)
17 frames = vec2frames(vector, Nw, Ns);
18
19 % Magnitude spectrum computation (as column vectors)
20 %MAG = abs(fft(frames,nfft,1));
21
22 m=size(frames);
23 t=1:m(2);
24 n=1:m(1);
25 n=n';
26
27 Zt=zeros(1,m(2));
28
29 for i=2:m(1)
30     m1=frames(i,:);
31     m2=frames(i-1,:);
32     s1=signo(m1);
33     s2=signo(m2);
34     Zt(i,:)=abs(s1-s2);

```



```

35 end
36
37 Zt_sum=1/2*(sum(Zt));
38
39 end
40
41 function s = signo(x)
42 s=x>0;
43 end

```

Listing 21: Cruce por cero

```

1 % Copyright (C) Ingenieria Electronica. 2018 Ucundinamarca
2 % Calcula el flujo espectral de @var{vector}.
3 % Author: Sebastian Munoz <sebastianmg.info@gmail.com>
4 % Created: 2016
5 % modified by: Jose Zamora <learningtesis@hotmail.com>
6 % Date: 2018
7 function [Ft_sum] = flujo(Tw, Ts, fs, vector)
8
9 Nw = round( 1E-3*Tw*fs ); % duracion del marco en muestras
10 Ns = round( 1E-3*Ts*fs ); % duracion del cambio de marco en muestras
11 nfft = 2^nextpow2( Nw ); % Longitud del analisis FFT
12
13 % Enmarcado y ventanamiento (marcos como columnas)
14 frames = vec2frames(vector, Nw, Ns);
15 % Calculo del espectro de magnitud (como vectores columna)
16 MAG = abs(fft(frames,nfft,1));
17
18 m=size(MAG);
19 t=1:m(2);
20 n=1:m(1);
21 n=n';
22
23 Ft=zeros(m);
24 mayor=max(MAG);
25
26 for i=2:m(2)
27     Nt_1=(MAG(:,i))/mayor(i); % se normaliza con el mismo numero???
28     Nt_2=(MAG(:,i-1))/mayor(i-1);
29     Ft(:,i)=(Nt_1-Nt_2).^2; % Flujo Espectral
30 end
31
32 Ft_sum=sum(Ft);
33
34 end

```

Listing 22: Flujo espectral

```

1 Tw = 25; % Duracion de la trama de analisis (ms)
2 Ts = 10; % Cambio de trama de analisis (ms)
3 t1= 90;
4 t2= 91;
5
6 for i=1:4
7     if (i==1)
8         wav_file =glob('/home/jestebanza/Aprendizaje/audicion computacional/
9         canciones/bambuco/*.wav');
10        file1 =glob('/home/jestebanza/Aprendizaje/audicion computacional/
11        canciones/bambuco/*.wav');
12        w=length(wav_file);
13        vec_caract_bam=zeros(10,w*98);
14    elseif (i==2)
15        wav_file =glob('/home/jestebanza/Aprendizaje/audicion computacional/
16        canciones/currulao/*.wav');
17        file2 =glob('/home/jestebanza/Aprendizaje/audicion computacional/
18        canciones/currulao/*.wav');
19        w=length(wav_file);
20        vec_caract_cur=zeros(10,w*98);
21    elseif (i==3)
22        wav_file =glob('/home/jestebanza/Aprendizaje/audicion computacional/
23        canciones/joropo/*.wav');
24        file3 =glob('/home/jestebanza/Aprendizaje/audicion computacional/
25        canciones/joropo/*.wav');
26        w=length(wav_file);
27        vec_caract_jor=zeros(10,w*98);
28    elseif (i==4)
29        wav_file =glob('/home/jestebanza/Aprendizaje/audicion computacional/
30        canciones/vallenato/*.wav');
31        file4 =glob('/home/jestebanza/Aprendizaje/audicion computacional/
32        canciones/vallenato/*.wav');
33        w=length(wav_file);
34        vec_caract_va=zeros(10,w*98);
35    end
36
37    for canciones=1:w
38        wav=wav_file(canciones,1);
39        wav=char(wav);
40        vec_caract = vector_tex_tim(wav,t=[t1 t2],Tw,Ts);
41
42        if (i==1)
43            vec_caract_bam =[vec_caract];
44        vec_caract_bam=vec_caract_bam';
45        long=size(vec_caract_bam)(1);
46    for canc= 1:long

```

```

39     printf("%0.10f,",vec_caract_bam(canc,:));
40     printf("%s",char(wav_file(canciones,1)));
41     printf("%s", ",bambuco");
42     printf("\n");
43 end
44     elseif (i==2)
45         vec_caract_cur=[vec_caract];
46         vec_caract_cur=vec_caract_cur';
47         long=size(vec_caract_cur)(1);
48     for canc= 1:long
49         printf("%0.10f,",vec_caract_cur(canc,:));
50         printf("%s",char(wav_file(canciones,1)));
51         printf("%s", ",currulao");
52         printf("\n");
53     end
54     elseif (i==3)
55         vec_caract_jor=[vec_caract];
56         vec_caract_jor=vec_caract_jor';
57         long=size(vec_caract_jor)(1);
58     for canc= 1:long
59         printf("%0.10f,",vec_caract_jor(canc,:));
60         printf("%s",char(wav_file(canciones,1)));
61         printf("%s", ",joropo");
62         printf("\n");
63     end
64     elseif (i==4)
65         vec_caract_va=[vec_caract];
66         vec_caract_va=vec_caract_va';
67         long=size(vec_caract_va)(1);
68     for canc= 1:long
69         printf("%0.10f,",vec_caract_va(canc,:));
70         printf("%s",char(wav_file(canciones,1)));
71         printf("%s", ",vallenato");
72         printf("\n");
73     end
74     end
75
76     end
77 end
78
79
80 csvwrite('textura_timbrica_bam.csv',[vec_caract_bam]);
81 csvwrite('textura_timbrica_cur.csv',[vec_caract_cur]);
82 csvwrite('textura_timbrica_jor.csv',[vec_caract_jor]);
83 csvwrite('textura_timbrica_va.csv',[vec_caract_va]);
84 csvwrite('textura_timbrica.csv',[vec_caract_bam;vec_caract_cur;

```

```

    vec_caract_jor;vec_caract_va]);
85 cell2csv('nombres.csv',[file1;file2;file3;file4]);

```

Listing 23: Textura timbrica

```

1
2 function vec_caract = vector_tex_tim(nombre,t=[x1 x2],Tw,Ts)
3 if !ischar(nombre)
4     error('no es un nombre de archivo');
5 end
6
7
8 alpha = 0.97;           % coeficiente de pre-enfasis
9 M = 20;                 % Cantidad de canales de filtros
10 C = 12;                 % Numero de coeficientes cepstrales
11 L = 22;                 % Parametro cepstral sine lifter
12 LF = 300;               % Limite de frecuencia inferior (Hz)
13 HF = 3700;              % Limite de frecuencia superior (Hz)
14
15     [vector,fs,nbits]=wavread(nombre,100); % lectura del audio para
    saber la frecuencia fs
16     [vector,fs,nbits]=wavread(nombre,t*fs);
17     vector=vector(:,1);
18
19     [Ct]=centroide(Tw,Ts,fs,vector);
20     [Ae]=atenuacion(Tw,Ts,fs,vector);
21     [Ft]=flujo(Tw,Ts,fs,vector);
22     [Zt]=cruce(Tw,Ts,fs,vector);
23     [MFCCs,FBEs,frames]=mfcc(vector,fs,Tw,Ts,alpha,@hamming,[LF HF],M,C
    +1,L);
24     MFCCs5=MFCCs(1:5,:);
25     L_E = low_e(Tw, Ts, fs, vector);
26     vec_caract =[Ct;Ae;Ft;Zt;MFCCs5];
27
28 end

```

Listing 24: Vector textura timbrica

B.2. R

```

1 # Copyright (C) Ingenieria Electronica. 2018 UCundinamarca
2 #
3 # Algoritmo k vecinos mas cercanos
4
5 # Author: Jose Zamora <learningtesis@hotmail.com>
6 # Created: 2018
7

```

```

8
9 library(ggplot2)
10 library(dplyr)
11 library(tidyr)
12 library(purrr)
13 library(readr)
14 library(ISLR)
15 library(knitr)
16 library(tidyr)
17 library(kknn)
18 library(BBmisc)
19 library(MASS)
20 library(e1071)
21 library(caret)
22 library(AppliedPredictiveModeling)
23
24 generos <- read.csv("modelo.csv")
25 canciones <- read.csv("canciones.csv")
26
27 generos <- normalize(generos, method = "range", range = c(0, 1), margin = 1
  L, on.constant = "quiet")
28 canciones <- normalize(canciones, method = "range", range = c(0, 1), margin
  = 1L, on.constant = "quiet")
29 lista <- subset(canciones, select = -nombre)
30
31 # 1 grupo
32 train <- subset(generos, select = -nombre)
33 trainx <- train[,1:9]
34 trainy <- train[,10]
35
36 teste <- rbind(canciones[1:98,])
37 teste <- rbind(teste, canciones[8528:8625,])
38 teste <- rbind(teste, canciones[17054:17151,])
39 teste <- rbind(teste, canciones[25482:25579,])
40
41 testa <- rbind(lista[1:98,])
42 testa <- rbind(testa, lista[8528:8625,])
43 testa <- rbind(testa, lista[17054:17151,])
44 testa <- rbind(testa, lista[25482:25579,])
45
46 set.seed(392)
47 control <- trainControl(method="repeatedcv", number=10, repeats=3)
48 modelo <- train(Gene~., data=train, method="knn", trControl=control,
  preProcess= c("center", "scale"), tuneLength=20)
49 test <- testa
50 testx <- test[,1:9]

```

```

51 testy <- test[,10]
52 predic <- predict(modelo, testx)
53 prop.table(table(predic))*100
54 confusionMatrix(predic, testy)
55
56 generos <- rbind(generos, teste)
57 testa <- 0
58 teste <- 0

```

Listing 25: Algoritmo de aprendizaje k-NN

```

1 # Copyright (C) Ingenieria Electronica. 2018 UCundinamarca
2 #
3 # Algoritmo Clasificador bayesiano ingenuo
4
5 # Author: Jose Zamora <learningtesis@hotmail.com>
6 # Created: 2018
7
8
9 library(ggplot2)
10 library(dplyr)
11 library(tidyr)
12 library(purrr)
13 library(readr)
14 library(ISLR)
15 library(knitr)
16 library(tidyr)
17 library(kknn)
18 library(BBmisc)
19 library(MASS)
20 library(e1071)
21 library(caret)
22 library(klaR)
23
24 generos <- read.csv("modelo.csv")
25 canciones <- read.csv("canciones.csv")
26
27 generos <- normalize(generos, method = "range", range = c(0, 1), margin = 1
  L, on.constant = "quiet")
28 canciones <- normalize(canciones, method = "range", range = c(0, 1), margin
  = 1L, on.constant = "quiet")
29 lista <- subset(canciones, select = -nombre)
30 grid <- data.frame(fL=c(0,0.5,1.0), usekernel = TRUE, adjust=c(0,0.5,1.0))
31 # 1 grupo
32 train <- subset(generos, select = -nombre)
33 trainx <- train[,1:9]
34 trainy <- train[,10]

```

```

35
36 teste <- rbind(canciones[1:98,])
37 teste <- rbind(teste, canciones[8528:8625,])
38 teste <- rbind(teste, canciones[17054:17151,])
39 teste <- rbind(teste, canciones[25482:25579,])
40
41 testa <- rbind(lista[1:98,])
42 testa <- rbind(testa, lista[8528:8625,])
43 testa <- rbind(testa, lista[17054:17151,])
44 testa <- rbind(testa, lista[25482:25579,])
45
46 set.seed(392)
47 control <- trainControl(method="repeatedcv", number=10, repeats=3)
48 modelo <- train(Gene~., data=train, method="nb", trControl=control,
  preProcess= c("center", "scale"), tuneGrid = grid)
49 test <- testa
50 testx <- test[,1:9]
51 testy <- test[,10]
52 predic <- predict(modelo, testx)
53 prop.table(table(predic))*100
54 confusionMatrix(predic, testy)
55
56 generos <- rbind(generos, teste)
57 testa <- 0
58 teste <- 0

```

Listing 26: Algoritmo de aprendizaje Clasificador Bayesiano ingenuo

```

1 # Copyright (C) Ingenieria Electronica. 2018 UCundinamarca
2 #
3 # Algoritmo vector soporte de maquina
4
5 # Author: Jose Zamora <learningtesis@hotmail.com>
6 # Created: 2018
7
8 library(ggplot2)
9 library(dplyr)
10 library(tidyr)
11 library(purrr)
12 library(readr)
13 library(ISLR)
14 library(knitr)
15 library(tidyr)
16 library(kknn)
17 library(BBmisc)
18 library(MASS)
19 library(e1071)

```

```

20 library(caret)
21 library(kernlab)
22
23 generos <- read.csv("modelo.csv")
24 canciones <- read.csv("canciones.csv")
25
26 generos <- normalize(generos, method = "range", range = c(0, 1), margin = 1
  L, on.constant = "quiet")
27 canciones <- normalize(canciones, method = "range", range = c(0, 1), margin
  = 1L, on.constant = "quiet")
28 lista <- subset(canciones, select = -nombre)
29 grid <- expand.grid(sigma = c(.01, .015, 0.2), C = c(0.75, 0.9, 1, 1.1,
  1.25))
30
31 # 1 grupo
32 train <- subset(generos, select = -nombre)
33 trainx <- train[,1:9]
34 trainy <- train[,10]
35
36 teste <- rbind(canciones[1:98,])
37 teste <- rbind(teste, canciones[8528:8625,])
38 teste <- rbind(teste, canciones[17054:17151,])
39 teste <- rbind(teste, canciones[25482:25579,])
40
41 testa <- rbind(lista[1:98,])
42 testa <- rbind(testa, lista[8528:8625,])
43 testa <- rbind(testa, lista[17054:17151,])
44 testa <- rbind(testa, lista[25482:25579,])
45
46 set.seed(392)
47 control <- trainControl(method="repeatedcv", number=10, repeats=3)
48 modelo <- train(Gene~., data=train, method="svmRadial", trControl=control,
  preProcess= c("center", "scale"), tuneGrid=grid, tuneLength=10)
49 test <- testa
50 testx <- test[,1:9]
51 testy <- test[,10]
52 predic <- predict(modelo, testx)
53 prop.table(table(predic))*100
54 confusionMatrix(predic, testy)
55
56 generos <- rbind(generos, teste)
57 testa <- 0
58 teste <- 0

```

Listing 27: Algoritmo de aprendizaje SVM

C. Tablas de resultado de clasificación

Titulo	Artista	Porcentaje de acierto
Yo era un niño	Diver Higuita	53,06124
On tabas	Edmundo Arias Y Su Con- junto	100
Tálamo de rosas	Espinosa Y Bedoya	100
Arroyito campesino	Estudiantina Colombia	100
Gloria	Estudiantina Colombia	61,85944
Ojos de yo no se que	Estudiantina Colombia	51,0204
Se que me quieres	Estudiantina Colombia	100
On tabas	Eugenio Ariza	98,9796
Soy colombiano	Garzón Y Collazos Con Jaime Llano González	96,93876
Prefiero no verte	Ginette Acevedo	100
Soy Caqueteño	Grupo Vocal Mixto Tayarú - Eduardo Bahamón H	60,20408
Cantar del río	Gustavo Motta	100
Fuego en el alma	Gómez Y Villegas	100
Bambuco grancolombiano	Intérprete Desconocido	96,938776
Antioqueña	Jaime Llano González	61,22448
El solterón	Jaime Llano González	67,34692
María Antonia	Jaime Llano González	61,22448
El mío	Jaime Llano González	100
Vivan las fiestas	Jaime Llano González	52,0408
Misia dolores	Jaime Martínez	95,91836
Tunjuelito	Jorge Ariza	64,29572
El corazón de la caña	José A. Morales	45,91836
Ayer me echaron del pue- blo	José A. Morales	100
El pañuelito	Julio Erazo Y Las Teresitas	89,79592
Lejanos ojos	Las Hermanas Garavito	60,20408
Niño divino	Las Hermanas Garavito	100
Cabañera	Los Cuatro Amigos	92,87516
Princesa imperial	Los Giraldinos	84,69388
La medallita	Los Hermanos Calero	76,5306
Mi cafetal	Los Hermanos Martínez	48,9796
Rosalinda	Los Hermanos Martínez	75,510204
Cuando contemplo tu boca	Los Hermanos Ríos	78,57144
Yo guardaré tu amor	Los Hermanos Ríos	89,79592
Cautivo	Los Luceros De Oiba	31,632652
Hermoso anhelo	Los Luceros De Oiba	60,20408

Tabla 15: Nombres muestras musicales del Bambuco en la clasificación

Titulo	Artista	Porcentaje de acierto
Se oye sona	Agrupación Changó	60,20408
Tápalo	Agrupación Changó	100
Como repicaba esa marimba	Ines Granja	70,40816
La churumbela	Canalon De Timbiquí	100
Adiós margarita	Canalon De Timbiquí	100
Arrullando	Canalon De Timbiquí	89,79592
Canto A mamá	Canalon De Timbiquí	83,67348
El curruco	Canalon De Timbiquí	100
El duende	Canalon De Timbiquí	93,87756
El gallo blanco	Canalon De Timbiquí	58,16328
Este niño quiere monte en el reloj	Canalon De Timbiquí	73,4694
Feliz cumpleaños	Canalon De Timbiquí	91,83672
Garcita morena	Canalon De Timbiquí	63,26532
La canción del camino	Canalon De Timbiquí	35,714284
La Juana	Canalon De Timbiquí	100
Lancherito	Canalon De Timbiquí	100
Mi canaleta	Canalon De Timbiquí	100
Molino mi molinete	Canalon De Timbiquí	18,367348
Qué pena	Canalon De Timbiquí	76,5306
Rio timbiquí	Canalon De Timbiquí	100
Tolero	Canalon De Timbiquí	67,34692
Tía tortuga	Canalon De Timbiquí	100
Vamos pastores Vamos	Canalon De Timbiquí	100
Zapato al pie	Canalon De Timbiquí	100
Chambacú	Totó la Momposina	29,591836
El negro caliente	Interprete desconocido	78,57144
Que pena	Canalon De Timbiquí	100
Sagrada santa maria	Canalon De Timbiquí	43,87756
Kilele	Grupo Bahia	100
Andarele	Grupo Bahía	69,38776
El birimbí	Grupo Bahía	100
Te vengo a cantar	Grupo Bahía	100
Volando	Grupo Bahía	100
Mi buenaventura	Grupo Bahía	33,673468
Brujería del bum	Grupo Buscajá	87,9592

Tabla 16: Nombres muestras musicales del Currulao en la clasificación

Titulo	Artista	Porcentaje de acierto
El colector	Cholo Valderrama	88,77552
Colombia	Cholo Valderrama	88,77552
Como quisiera olvidarte	Armando Martinez-Iván José	100
Como no voy a decir	Luis Silva	44,89796
De que tamaño es tu amor	Reynaldo armas	100
Desesperacion	Aries vigoth	100
El pajarillo	Denny del Rio	100
Yo soy del llano	Elda flores	100
El criollo sortario	Interprete desconocido	100
El indio	Reinaldo Armas	100
Enamorado de ti	Interprete desconocido	86,73468
Fiesta en Elorza	Eneas Perdomo	82,65308
El tigre de payara	Francisco Montoya	69,38776
Hoy por fin	Reinaldo Armas	100
Ironia	musica llanera	100
El sifrino y el veguero	Jorge guerrero	100
Joropo libre	Hussein Hernández y Leidy Cachay	100
Joropo	Jhon Aguirre y Katherine Gonzalez	100
Lucerito	Reinaldo Armas	100
Ella o el	Luis Silva	97,9592
Enfurecida	Luis Silva	87,75512
Farsante	Luis Silva	83,67348
Scarlet	Luis Silva	45,91836
La muerte del rucio moro	Reinaldo Armas	100
Llanerazo	Interprete desconocido	100
Enfurecida	Luis Silva	64,28572
Atrapado	Luis Silva	59,18368
La más bonita	Luis Silva	58,16328
Botaloneao	Luis Silva	55,10204
Llamarada	Luis Silva	100
Pajarillo	Luis Silva	24,489796
Piel canela	Luis Silva	64,28572
Mi amor llanero	Elda florez	84,69388
Mujer dejate querer	Armando Martinez	100
Métele candela al rancho	Reinaldo Armas	76,5306

Tabla 17: Nombres muestras musicales del Joropo en la clasificación

Titulo	Artista	Porcentaje de acierto
De niña a mujer	Los Chiches Del Vallenato	100
Despues de tantos años	Los Diablitos	85,71428
Después de amar	Los Inquietos Del Vallenato	94,89796
Después de ti	Los Inquietos Del Vallenato	100
Dicen que soy	Los Muchachos	98,3148
Difícil de igualar	Binomio De Oro	100
Dime como te olvido	Binomio De Oro	84,69388
Dime dime	Los Valbuena	63,26532
Distintos destinos	Binomio De Oro	91,83672
Donde esta	Los Diablitos	73,4694
Déjala	Los Diablitos	100
El chanchullito	Diomedez Diaz	96,93876
Ella tiene todo	Fabián Corrales	100
Ella era todo	Los Inquietos Del Vallenato	100
Eras diferente	La Decisión Vallenata	98,9796
Eres mi vida	La Tribu Vallenata	63,26532
Es dolor	Omar Geles	100
Estas en mi	Luis Mateus	100
Fabula de amor	Los Chiches Vallenatos	100
Hasta cuando señor	Los Diablitos	98,9796
Lejos de ti	Binomio De Oro	100
Llegaste tu	Adriana Lucia	100
Llévame en tus sueños	Jorge Celedón	100
Lo mismo me da	Diomedez Diaz	76,5306
Magico sagitario	Interprete desconocido	100
Me canse de usted	Las Diosas Del Vallenato	100
Me vas a extrañar	Binomio De Oro	100
Mi Unica Verdad	Jorge Celedón	100
Mi mejor sueño	Ilusión Vallenata	100
Mi novia mujer	Los Diablitos	100
Mi presidio	Romualdo Brito	100
Muero por verla	Luis Mateus	100
No podrán separarnos	Jorge Celedón	92,85716
No debemos terminar	Los Emigrantes	100
No he podido ser feliz	Los Gigantes Del Vallenato	100

Tabla 18: Nombres muestras musicales del Vallenato en la clasificación