



Trabajo Fin de Máster

Máster en Ciencia de Datos e Ingeniería de Datos en la Nube

Clasificación Automática de Géneros Musicales

Autor: Daniel Coronado Martín

Tutor: María Julia Flores Gallego

Julio, 2023

A mis padres

Declaración de Autoría

Yo, Daniel Coronado Martín con DNI 47400140S, declaro que soy el único autor del trabajo fin de grado titulado “Clasificación Automática de Géneros Musicales” y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Albacete, a 14 de julio de 2023

Fdo: Daniel Coronado Martín

Resumen

El presente trabajo de investigación aborda el desafío de la clasificación de géneros musicales en un contexto cada vez más diverso y dinámico. En un mundo donde la música es una parte esencial de nuestra vida cotidiana y se produce una multiplicidad de estilos y fusiones, comprender y categorizar los diversos géneros musicales se ha vuelto fundamental para una amplia gama de aplicaciones, como la recomendación de música personalizada, la curación de listas de reproducción, la promoción de artistas emergentes y la exploración de nuevas tendencias en la industria musical.

La clasificación de géneros musicales presenta un desafío único debido a la naturaleza subjetiva y multidimensional de la música. Cada género musical posee características distintivas en términos de ritmo, armonía, instrumentación, estructura y emociones transmitidas. Por lo tanto, es necesario emplear enfoques analíticos y de aprendizaje automático sofisticados para capturar la complejidad y la diversidad de la música en sus diversas formas.

Este estudio propone un enfoque basado en técnicas de aprendizaje automático para clasificar automáticamente canciones en diferentes géneros musicales. Se recopiló una extensa base de datos de canciones de diversos géneros y se extrajeron características acústicas relevantes, como el ritmo, el timbre y la energía. Estas características se utilizaron para entrenar y evaluar varios algoritmos de aprendizaje automático, como árboles de decisión, redes convolucionales y redes recurrentes.

Los resultados experimentales mostraron una precisión significativa en la clasificación de géneros musicales. Se exploraron técnicas de procesamiento de lenguaje natural para extraer características relevantes de las letras y se encontró que la combinación de características lingüísticas y acústicas mejora aún más el rendimiento de la clasificación.

Agradecimientos

Quiero expresar mi profundo agradecimiento a las personas que han sido fundamentales en la realización de mi TFM. En primer lugar, me gustaría dedicar mis agradecimientos a mi tutora, Julia, cuya orientación y apoyo fueron indispensables para sacar adelante este proyecto.

También quiero agradecer a mis compañeros y profesores, Miguel Ángel Cantero, Fernando Rubio y Juan Carlos Alfaro, por su colaboración y contribución al desarrollo de mi trabajo. Su aportación y feedback fueron invaluableles en cada etapa del proceso.

Además, no puedo dejar de mencionar a mis padres y a mi hermano, quienes siempre han estado a mi lado brindándome su apoyo incondicional. Su amor y aliento han sido una fuente constante de motivación para mí.

Por último, quiero hacer una mención especial a José Antonio Gámez, mi profesor de Sistemas Inteligentes durante mi tercer año de carrera. Fue gracias a su pasión y dedicación por el tema que descubrí mi verdadera pasión por la Inteligencia Artificial. A partir de ese momento, he seguido aprendiendo y formándome en este campo sin descanso. No puedo evitar preguntarme cómo habría sido mi camino si no hubiera tenido la oportunidad de cursar su asignatura.

A todas estas personas, mi más sincero agradecimiento. Sin su ayuda, guía y apoyo, este TFM no habría sido posible.

Índice general

Capítulo 1	Introducción	1
Capítulo 2	Estado del Arte	5
2.1	Motivación	5
2.2	Aprendizaje en música mediante imágenes (espectogramas)	6
2.3	Técnicas que emplearemos en este trabajo	7
2.3.1	Árboles de Decisión	8
2.3.2	Random Forest	9
2.3.3	Redes neuronales convoluciones (CNN)	10
2.3.4	LSTM	12
2.4	Construcción de modelos híbridos o combinados	14
2.4.1	Mediante media aritmética	14
2.4.2	Mediante Model Stacking	15
2.4.3	Empleando Chain Classifier	16
2.5	Consideraciones finales sobre el problema	17
Capítulo 3	Caso de Estudio	18
3.1	Introducción	18
3.2	Revisión de Bases de Datos	19
3.3	Enriquecimiento de la Base de Datos	26
Capítulo 4	Experimentación	30
4.1	Análisis Exploratorio de Datos	30

4.2	Machine Learning	38
4.3	Deep Learning	41
4.4	Ensemble	50
Capítulo 5	Conclusiones y Trabajo Futuro	56
5.1	Conclusiones	56
5.2	Trabajo futuro	58
Bibliografía		63
Anexo I.	Desarrollo Realizado	66
I.1	Enriquecimiento de la Base de Datos	66
I.2	Machine Learning	66
I.3	Deep Learning	66
I.4	Ensemble	66

Índice de figuras

Figura 2.1. Diagrama Árbol de Decisión.....	8
Figura 2.2. Diagrama Redes Convolucionales Unidimensionales	11
Figura 2.3. Diagrama Redes de Memoria a Corto Plazo	13
Figura 3.1. Spotify Million Song Dataset	20
Figura 3.2. 5 Million Song Lyrics Dataset	23
Figura 3.3. Flujo de extracción de características de audio	28
Figura 3.4. SMS tras el enriquecimiento con características de audio	29
Figura 4.1. Distribución de géneros musicales	31
Figura 4.2. Gráfico de dispersión Energía / Valencia	32
Figura 4.3. Distribución de Energía por Género	33
Figura 4.4. Distribución de Speechiness por Género	34
Figura 4.5. Casos curiosos Speechiness	36
Figura 4.6. Wordclouds por género	37
Figura 4.7. Distribución final Train y Test	37
Figura 4.8. Diagrama Redes Convolucionales Unidimensionales - 2	43
Figura 4.9. Arquitectura Keras Red Convolucional	44
Figura 4.10. Código Callback Learning Rate Adaptativo	46
Figura 4.11. Gráfico de evolución Red Convolucional	46
Figura 4.12. Arquitectura Keras Red de Memoria a Corto Plazo	47
Figura 4.14. Gráfico de evolución Red de Memoria a Corto Plazo	48
Figura 4.15. Predicciones individuales Decision Tree	50

Índice de tablas

Tabla 4.1. Resultados Machine Learning	41
Tabla 4.2. Resultados Deep Learning	49
Tabla 4.3. Resultados Ensemble por Media Aritmética	51
Tabla 4.4. Resultados Ensemble por Model Stacking	53
Tabla 4.5. Resultados Ensemble por Chain Classifier	54

Capítulo 1

Introducción

Este estudio se enfoca en analizar las letras y las características auditivas de las canciones con el fin de clasificar los géneros musicales. El objetivo principal es desarrollar un enfoque científico y objetivo que permita identificar de manera precisa y automatizada los géneros musicales utilizando estas dos fuentes de información.

El estudio de la clasificación de géneros musicales es de gran relevancia en diversos ámbitos, como la industria discográfica, la recomendación de música y la comprensión cultural de las obras musicales. Sin embargo, esta tarea ha sido históricamente desafiante debido a la falta de criterios objetivos y consistentes para definir los géneros. Las clasificaciones tradicionales han dependido de la percepción individual de los oyentes o de la opinión de expertos en música, lo que ha generado discrepancias y ambigüedades en su análisis y estudio.

El avance en el campo de la inteligencia artificial y el procesamiento del lenguaje natural ha brindado nuevas oportunidades para abordar este problema de manera más rigurosa y objetiva. La disponibilidad de grandes cantidades de datos musicales y técnicas de aprendizaje automático ha permitido explorar en profundidad las relaciones entre las características musicales y los géneros asociados. Al combinar el análisis de las letras con las características auditivas de las canciones, se espera obtener una clasificación más precisa y completa de los géneros musicales.

Los objetivos específicos de este trabajo son:

1. Investigar y seleccionar las técnicas de procesamiento de lenguaje natural más adecuadas para el análisis de las letras de las canciones, considerando su estructura, contenido semántico y características lingüísticas.
2. Explorar las características auditivas relevantes en la música para la clasificación de géneros, incluyendo aspectos como el ritmo, la melodía, la instrumentación y otros atributos musicales.
3. Desarrollar un modelo de clasificación que integre tanto el análisis de las letras como las características auditivas, utilizando técnicas de aprendizaje automático y minería de datos.
4. Evaluar y validar el modelo propuesto utilizando conjuntos de datos amplios y diversos, con el fin de determinar su eficacia y generalidad en la clasificación de géneros musicales.
5. Contribuir al avance en el campo de la clasificación de géneros musicales y sentar las bases para futuras investigaciones en la materia, fomentando la aplicación de enfoques científicos y tecnológicos en el análisis y comprensión de la música.

Se espera que los resultados de esta investigación contribuyan al avance de la clasificación automática de géneros musicales, proporcionando una herramienta útil para diversas aplicaciones en la industria musical y el ámbito académico.

El presente documento se organiza de la siguiente manera para abordar de manera exhaustiva la investigación sobre la clasificación de géneros musicales a partir de las letras y las características auditivas:

En el Capítulo 2, se realiza una revisión exhaustiva de la literatura relacionada con la clasificación de géneros musicales y los enfoques existentes para su análisis. Se exploran los diferentes criterios utilizados históricamente para la clasificación de géneros, así como los desafíos y limitaciones asociados. Además, se revisan los avances en el campo de la inteligencia artificial, el procesamiento del lenguaje natural y el análisis de señales de audio que han permitido abordar este problema desde una perspectiva más científica y objetiva.

En el Capítulo 3, se detalla la metodología propuesta para la clasificación de géneros musicales a partir de las letras y las características auditivas. Se describen en detalle las técnicas de procesamiento de lenguaje natural seleccionadas para el análisis de las letras, así como las características auditivas relevantes consideradas en el estudio. Se explican también los criterios de selección de las canciones y las letras, así como los métodos empleados para recopilar y preprocesar los datos.

En el Capítulo 4, se presenta el diseño experimental junto con los algoritmos de aprendizaje automático utilizados para desarrollar el modelo de clasificación. Se exponen los resultados obtenidos a partir de la implementación de los modelos propuestos. Se analizan y discuten en detalle los resultados de la clasificación de géneros musicales, evaluando la precisión y el rendimiento general de los modelos. Se realizan comparaciones entre modelos discutiendo sus ventajas y limitaciones.

En el Capítulo 5, se extraen las conclusiones principales de la investigación y se discuten las implicaciones de los resultados obtenidos. Se resumen los hallazgos clave y se ofrecen recomendaciones para futuras investigaciones en el campo de la clasificación de géneros musicales a partir de las letras y las características auditivas.

Finalmente, en el Apéndice se incluyen los detalles técnicos adicionales sobre las herramientas y librerías utilizadas en la implementación del modelo, así como los fragmentos de código relevantes para su comprensión y replicación.

A través de esta estructura, se pretende ofrecer un enfoque completo y sistemático para abordar la problemática planteada, presentando los fundamentos teóricos, la metodología empleada, los resultados obtenidos y las conclusiones derivadas de la investigación realizada.

Capítulo 2

Estado del Arte

2.1 Motivación

La clasificación del género musical a través de las letras y las características auditivas de las canciones es un tema ampliamente investigado en el ámbito de la ciencia de datos y la inteligencia artificial. En la literatura existente, se han utilizado diferentes técnicas de aprendizaje automático para abordar este problema, incluyendo redes neuronales, árboles de decisión, SVM y Naive Bayes.

En cuanto a las características auditivas utilizadas para la clasificación, se han explorado diversas variables, como el tempo, la tonalidad, la acústica y otros parámetros relacionados con el audio. Por ejemplo, en un estudio de Tsaptsinos et al. (2016), se utilizó una combinación de características acústicas, incluyendo el tempo, la densidad espectral de energía y la entropía de la señal, para clasificar 10 géneros diferentes con una precisión de hasta el 71%.

Otro estudio interesante es el de Li et al. (2018), en el que se utilizó una combinación de características acústicas y de lenguaje natural, incluyendo la rima, la repetición de palabras y la densidad de vocabulario, para clasificar 5 géneros diferentes con una precisión de hasta el 83%.

En cuanto a las métricas de evaluación utilizadas para medir el desempeño de los modelos de clasificación, se han utilizado diferentes medidas, como la precisión, el recall, la F1-score y el área bajo la curva ROC. Por ejemplo, en un estudio de Han et al. (2020), se utilizó el área bajo la curva ROC para comparar el desempeño de diferentes modelos de clasificación en la tarea de clasificar 4 géneros musicales con características de audio y texto, obteniendo una precisión promedio del 84.3%.

Es importante mencionar que, en la tarea de clasificación de géneros musicales, existe el problema del *data leakage*, que puede afectar la capacidad de generalización de los modelos de clasificación. Para mitigar este problema, se han utilizado técnicas como la validación cruzada, que permite evaluar el desempeño del modelo en datos no vistos durante el entrenamiento. Por ejemplo, en un estudio de Cao et al. (2021), se utilizó una validación cruzada de 10 pliegues para evaluar el desempeño de un modelo de clasificación basado en redes neuronales en la tarea de clasificar 6 géneros diferentes con características de audio, obteniendo una precisión promedio del 79%.

2.2 Aprendizaje en música mediante imágenes (espectrogramas)

Además de las técnicas de aprendizaje automático tradicionales, recientemente se han explorado enfoques basados en imágenes para la clasificación de géneros musicales utilizando espectrogramas. Los espectrogramas son representaciones gráficas de la señal de audio que permiten visualizar la distribución de energía en diferentes frecuencias a lo largo del tiempo.

En la literatura existente, se han utilizado diferentes técnicas de clasificación de imágenes, como las redes neuronales convolucionales (CNN), para procesar los espectrogramas y clasificar los géneros musicales. Por ejemplo, en un estudio de Lee et al. (2017), se utilizó una CNN para extraer características de los espectrogramas y clasificar 6 géneros diferentes con una precisión de hasta el 75%.

Otro estudio interesante es el de Li et al. (2020), en el que se propuso una red neuronal recurrente (RNN) para clasificar los géneros musicales utilizando espectrogramas. En este trabajo, se utilizó una RNN bidireccional para capturar la relación temporal en los espectrogramas y una capa de atención para enfocar la red en las partes más importantes del espectrograma. Se obtuvo una precisión de hasta el 83% en la clasificación de 10 géneros diferentes.

Cabe mencionar que estas técnicas basadas en imágenes pueden ser útiles en la clasificación de géneros musicales, ya que permiten capturar información detallada sobre las características acústicas de las canciones. Además, el uso de espectrogramas como entrada puede ser una forma más robusta de procesar las señales de audio, ya que elimina la necesidad de preprocesamiento manual de las características de audio.

2.3 Técnicas que emplearemos en este trabajo

En este estudio se aborda la clasificación de géneros musicales a partir de la combinación de dos clasificadores con enfoques completamente distintos. Uno del ámbito del Machine Learning que infiere sobre características descriptivas de las canciones como la valencia, energía, duración, clave o tempo; y otro del ámbito del Procesamiento Natural del Lenguaje (NLP) que interpreta la relación existente entre palabras de una misma lyrics.

Para la parte de las características auditivas de las canciones se proponen dos modelos, Árboles de Decisión y Bosques Aleatorios.

Los Árboles de Decisión presentan una estructura basada en árboles dirigidos acíclicos. Cada nodo de este árbol dirigido acíclico corresponde con una característica o atributo del conjunto de datos. El nodo raíz del árbol representa la característica más importante, que se selecciona utilizando métricas como la ganancia de información o la impureza de Gini (Hastie, Tibshirani, & Friedman, 2009).

2.3.1 Árboles de Decisión

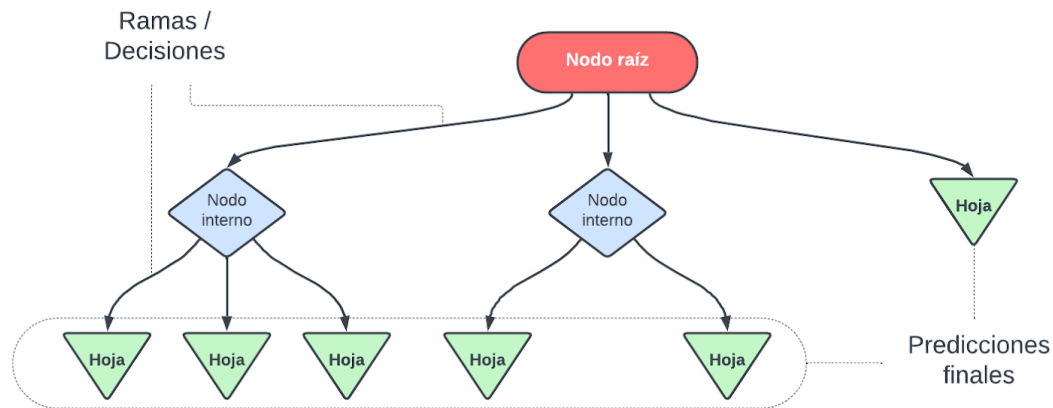


Figura 2.1. Diagrama Árbol de Decisión

A medida que se avanza en el árbol desde el nodo raíz hacia las hojas, se toman decisiones basadas en los valores de los atributos. Cada nodo interno representa una regla de decisión que divide el conjunto de datos en subconjuntos más pequeños en función de una condición sobre la característica correspondiente (Quinlan, 1986). Los nodos hoja, también conocidos como nodos terminales, representan las etiquetas de clasificación o los valores de regresión finales.

La construcción del árbol de decisión se realiza mediante algoritmos de partición recursiva como ID3, C4.5 o CART (Breiman, Friedman, Olshen, & Stone, 1984). Estos algoritmos evalúan diferentes atributos y sus divisiones óptimas en cada nivel del árbol, utilizando medidas como la entropía o la impureza de Gini para determinar la calidad de la división.

El proceso de entrenamiento del modelo de árbol de decisión implica ajustar los parámetros del árbol para maximizar su capacidad de generalización. Esto se logra mediante técnicas como la poda, que elimina subárboles innecesarios o sobreajustados, y la limitación de la profundidad del árbol (Mitchell, 1997).

Una vez construido y entrenado, el modelo de árbol de decisión puede utilizarse para clasificar nuevas instancias o predecir valores continuos. Esto se logra siguiendo el camino desde la raíz hasta una hoja, aplicando las reglas de decisión correspondientes en cada nodo.

En resumen, el modelo de árbol de decisión es una estructura jerárquica y direccionada que utiliza atributos de los datos para tomar decisiones basadas en reglas de división óptimas (Hastie, Tibshirani, & Friedman, 2009). Su construcción y entrenamiento se basan en algoritmos de partición recursiva, y una vez establecido, puede utilizarse para la clasificación o regresión de nuevas instancias.

2.3.2 Random Forest

El modelo de Random Forest (Bosque Aleatorio) es una técnica de aprendizaje automático que combina múltiples árboles de decisión para mejorar la precisión y la capacidad de generalización (Breiman, 2001). Su estructura se basa en un conjunto de árboles de decisión independientes, donde cada árbol es entrenado en una muestra aleatoria del conjunto de datos original.

Cada árbol en el Random Forest se construye utilizando un subconjunto aleatorio de características (atributos) seleccionadas del conjunto de datos completo. Esto se conoce como muestreo de características o selección de características aleatorias. La selección aleatoria de características ayuda a reducir la correlación entre los árboles y permite que cada árbol aporte una perspectiva diferente y única en la clasificación o regresión (Liaw & Wiener, 2002).

Durante el proceso de entrenamiento de un Random Forest, cada árbol se construye utilizando el algoritmo de árbol de decisión, como por ejemplo, ID3, C4.5 o CART. Cada árbol se entrena de manera independiente utilizando una técnica llamada bootstrapping, donde se genera una muestra aleatoria con reemplazo a partir del conjunto de datos original. Esto significa que algunas instancias pueden estar presentes múltiples veces en una muestra y otras pueden no estar presentes en absoluto.

Una vez que todos los árboles del Random Forest están contruidos, se utiliza un proceso de votación para tomar una decisión final de clasificación o regresión. En el caso de clasificación, cada árbol emite su voto para la clase predicha, y la clase con mayor número de votos se selecciona como la predicción final. En el caso de regresión, los resultados de los árboles se promedian para obtener una estimación final.

El Random Forest es especialmente efectivo para evitar el sobreajuste y mejorar la capacidad de generalización, ya que la combinación de múltiples árboles reduce el impacto de los errores individuales y mejora la robustez del modelo (Cutler et al., 2007). Además, el uso de muestreo aleatorio de características y bootstrapping ayuda a capturar la diversidad y la variabilidad en el conjunto de datos.

Por otro lado, para la parte de las lyrics se proponen dos modelos de Procesamiento Natural del Lenguaje, CNN 1D y LSTM.

2.3.3 Redes neuronales convoluciones (CNN)

Las redes neuronales convolucionales unidimensionales (CNN 1D) son un tipo de arquitectura de redes neuronales diseñadas específicamente para procesar datos secuenciales, como secuencias de texto, series de tiempo o señales unidimensionales. Estas redes están inspiradas en la organización del sistema visual en los seres vivos y han demostrado ser muy efectivas en una amplia gama de tareas de aprendizaje automático (Schmidhuber, 2015; LeCun, Bengio, & Hinton, 2015).

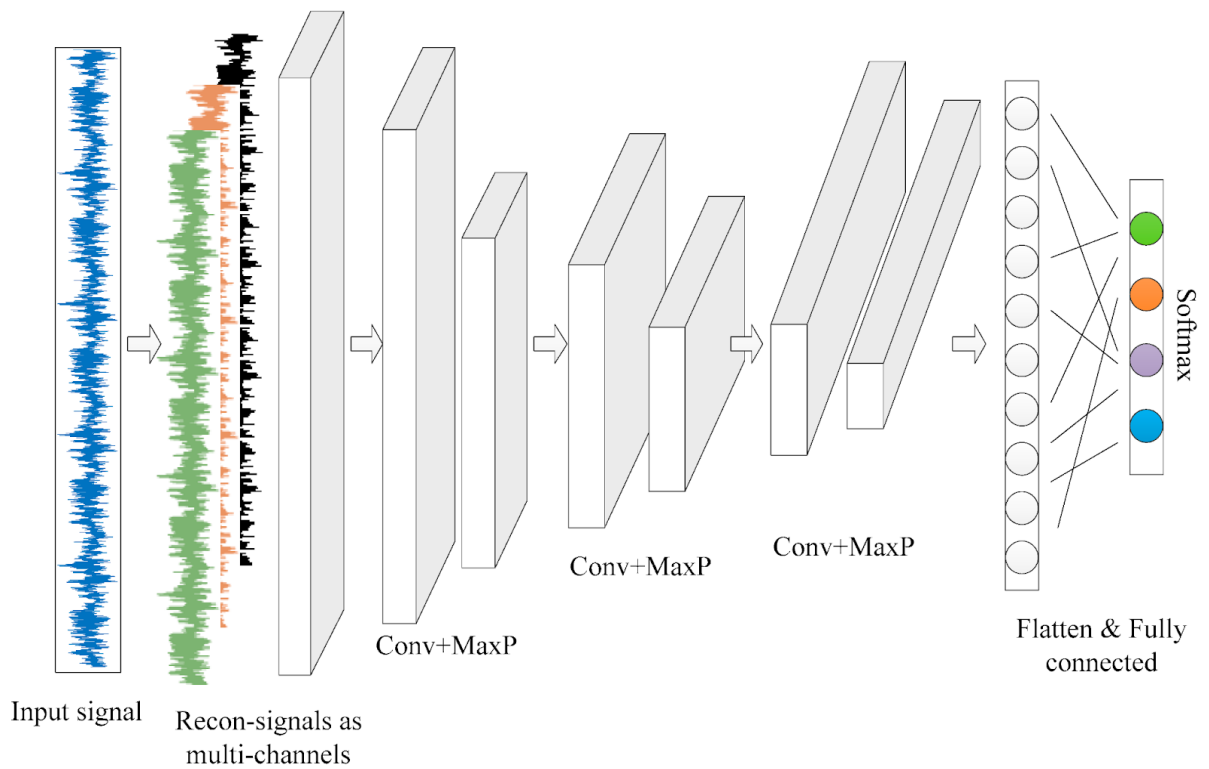


Figura 2.2. Diagrama Redes Convolucionales Unidimensionales

La estructura básica de una CNN 1D consiste en capas de convolución seguidas de capas de agrupación (pooling) y capas de activación no lineales. La convolución es el proceso central en una CNN 1D y se utiliza para extraer características relevantes de los datos secuenciales. Cada capa de convolución en una CNN 1D aplica un conjunto de filtros (también llamados kernels) a una ventana deslizante de la secuencia de entrada. Cada filtro detecta patrones locales en la secuencia, capturando características como bordes, formas o estructuras recurrentes. Durante la convolución, se calcula el producto punto entre los valores de la ventana y los pesos del filtro, generando un mapa de características que resalta la presencia de ciertos patrones en la secuencia (Bai, Kolter, & Koltun, 2018).

Después de la capa de convolución, se suele aplicar una función de activación no lineal, como la función ReLU (Rectified Linear Unit), para introducir la no linealidad en la red y permitir la representación de relaciones complejas entre los datos. La función ReLU transforma cualquier valor negativo a cero, manteniendo los valores positivos sin cambios.

A continuación, se aplica una capa de agrupación (pooling), que reduce la dimensionalidad espacial de las características extraídas y ayuda a capturar características invariantes a pequeñas variaciones en la ubicación. Una operación de agrupación común en una CNN 1D es el agrupamiento máximo (max pooling), que selecciona el valor máximo dentro de una ventana deslizante.

Estas capas de convolución y agrupación se pueden apilar en múltiples niveles, aumentando gradualmente la complejidad y abstracción de las características extraídas. Al final de la arquitectura, se suelen agregar capas totalmente conectadas (fully connected) para realizar la clasificación o la predicción final.

Durante el entrenamiento de una CNN 1D, se utilizan técnicas de aprendizaje supervisado, como la retropropagación del error, para ajustar los pesos de los filtros y las conexiones de la red con el objetivo de minimizar una función de pérdida, que mide la discrepancia entre las predicciones de la red y las etiquetas de los datos de entrenamiento (Kim, 2014).

En resumen, las redes neuronales convolucionales unidimensionales (CNN 1D) son una arquitectura de redes neuronales especialmente diseñada para procesar datos secuenciales. Utilizan capas de convolución, activaciones no lineales y agrupación para extraer características relevantes de la secuencia y lograr un aprendizaje efectivo en tareas como el procesamiento de texto, el análisis de series de tiempo y el procesamiento de señales unidimensionales (Sandler et al., 2018).

2.3.4 LSTM

Las redes de memoria a corto plazo LSTM (Long Short-Term Memory, por sus siglas en inglés) son un tipo de arquitectura de red neuronal recurrente (RNN) diseñada específicamente para superar las limitaciones de las RNN tradicionales en la tarea de modelado de secuencias y retención de información a largo plazo (Hochreiter & Schmidhuber, 1997).

En un nivel técnico, una LSTM consta de unidades de memoria llamadas celdas LSTM. Cada celda LSTM tiene tres puertas principales: la puerta de entrada (input gate), la puerta de olvido (forget gate) y la puerta de salida (output gate) (Graves, Mohamed & Hinton, 2013). Estas puertas son controladas por funciones de activación sigmoideas, que permiten controlar el flujo de información dentro de la celda.

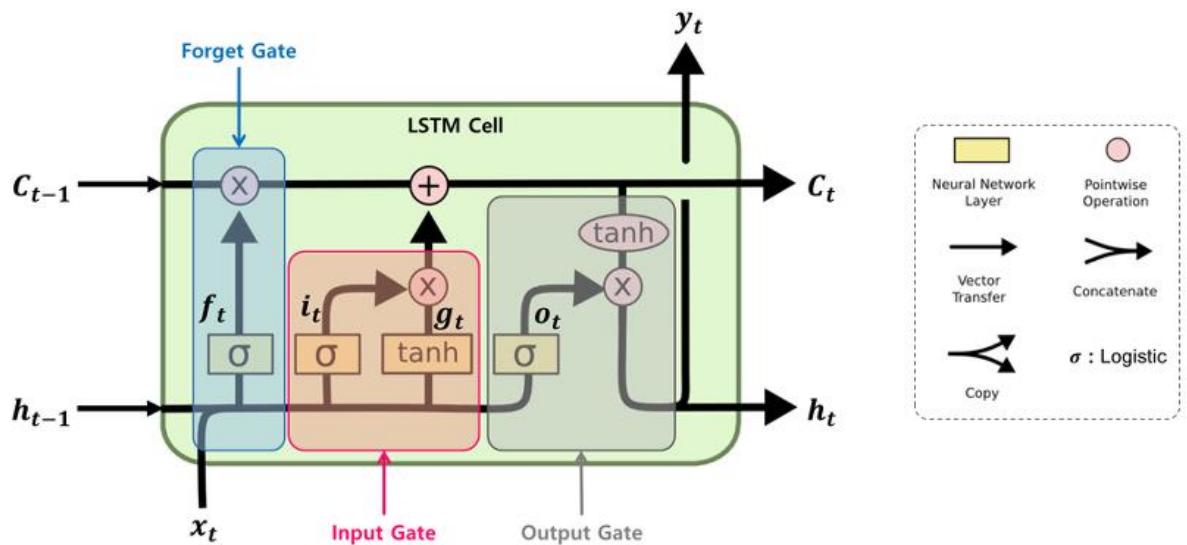


Figura 2.3. Diagrama Redes de Memoria a Corto Plazo

La puerta de entrada decide cuánta información nueva se debe agregar a la memoria actual. Utiliza la información de la entrada actual y la memoria anterior para calcular una actualización de la memoria propuesta (Graves, 2012).

La puerta de olvido decide qué información antigua debe descartarse de la memoria actual. Utiliza la información de la entrada actual y la memoria anterior para calcular un valor de olvido, que controla la cantidad de información antigua que se mantendrá (Gers, Schmidhuber & Cummins, 2000).

La puerta de salida determina qué información se debe transmitir como salida de la celda LSTM. Utiliza la información de la entrada actual y la memoria actualizada para calcular una salida basada en el estado actual de la memoria.

Además de las puertas, cada celda LSTM tiene una unidad de memoria interna que almacena y actualiza la información a lo largo del tiempo. Esta unidad de memoria permite a la LSTM retener información a largo plazo sin verse afectada negativamente por el problema del desvanecimiento o la explosión del gradiente que afecta a las RNN tradicionales (Greff, Srivastava, Koutník, Steunebrink & Schmidhuber, 2017).

La arquitectura LSTM se utiliza ampliamente en tareas de procesamiento de lenguaje natural, como el reconocimiento de voz, la traducción automática y la generación de texto. Su capacidad para modelar secuencias largas y capturar dependencias a largo plazo ha demostrado ser altamente efectiva en la resolución de este tipo de problemas.

2.4 Construcción de modelos híbridos o combinados

Para combinar las decisiones de los modelos de ambas partes se siguieron distintas técnicas de ensemble: Media Aritmética, Model Stacking y Chain Classifier.

2.4.1 Mediante media aritmética

El proceso de combinación mediante la media aritmética implica calcular el promedio de las predicciones generadas por los modelos individuales. Para ello, se suman todas las predicciones y se dividen por el número total de modelos considerados. Esta operación aritmética proporciona una única predicción combinada que representa el consenso entre los diferentes modelos.

Es importante destacar que este enfoque de combinación asume que todos los modelos contribuyen de manera igualmente válida a la predicción final y no se tiene en cuenta su rendimiento individual. En casos donde algunos modelos sean más confiables o precisos que otros, se pueden aplicar técnicas de ponderación para asignar pesos diferentes antes de calcular la media.

2.4.2 Mediante Model Stacking

Para el de tipo Model Stacking, se construyen múltiples modelos predictivos llamados "modelos base" o "modelos nivel 0". Estos modelos base pueden ser de diferentes tipos o utilizar diferentes algoritmos de aprendizaje automático. Cada modelo base se entrena con el conjunto de datos de entrenamiento y genera predicciones para el conjunto de datos de prueba.

Luego, las predicciones generadas por los modelos base se utilizan como características de entrada para un "modelo meta" o "modelo nivel 1". Este modelo meta se entrena con las características generadas y los valores reales del conjunto de datos de entrenamiento. Posteriormente, se utiliza para hacer predicciones en el conjunto de datos de prueba.

El objetivo principal del ensemble de Model Stacking es aprovechar las fortalezas de los diferentes modelos base, ya que cada modelo puede capturar diferentes patrones y relaciones en los datos. Al combinar las predicciones de estos modelos base en un modelo meta, se espera obtener una mayor capacidad de generalización y una mejora en la precisión predictiva.

Es importante destacar que el proceso de entrenamiento del ensemble de Model Stacking implica la división del conjunto de datos de entrenamiento en varias partes para entrenar los modelos base y el modelo meta. Además, se pueden aplicar técnicas como la validación cruzada para evaluar y ajustar el rendimiento del ensemble.

2.4.3 Empleando Chain Classifier

Y por último el ensemble de tipo Chain Classifier consiste en un conjunto de clasificadores individuales, cada uno de los cuales se entrena para predecir una etiqueta específica en función de las características de entrada y las predicciones de las etiquetas anteriores en la secuencia. En este contexto, las etiquetas anteriores actúan como características adicionales para el clasificador actual. Esto permite capturar las relaciones de dependencia entre las etiquetas y mejorar la precisión general del modelo.

Durante la fase de entrenamiento, se utiliza un enfoque iterativo para ajustar los clasificadores individuales en el conjunto. En cada iteración, se entrena un clasificador específico para predecir una etiqueta en función de las características de entrada y las predicciones de las etiquetas anteriores. Luego, las predicciones del clasificador actual se utilizan como características para entrenar el siguiente clasificador en la secuencia. Este proceso continúa hasta que se han entrenado todos los clasificadores.

Una vez que el Chain Classifier ha sido entrenado, se utiliza para predecir las etiquetas secuenciales de nuevas instancias de datos. Durante la inferencia, las características de entrada se pasan al primer clasificador en la secuencia, que realiza una predicción inicial. Luego, la predicción se utiliza como característica de entrada para el siguiente clasificador en la secuencia, y así sucesivamente, hasta que se obtienen las predicciones finales de todas las etiquetas en la secuencia.

2.5 Consideraciones finales sobre el problema

En general, la clasificación del género musical a través de las letras y las características auditivas de las canciones es un tema de investigación bien establecido en el ámbito de la ciencia de datos y la inteligencia artificial. La literatura existente ha utilizado diferentes técnicas de aprendizaje automático, características auditivas y métricas de evaluación para abordar este problema con resultados prometedores en la clasificación de varios géneros musicales.

En este estudio se aplicarán algunas de estas técnicas además de la combinación de otras ya existentes, comparándolas en términos de accuracy y tiempo de ejecución.

Capítulo 3

Caso de Estudio

3.1 Introducción

La elección de una base de datos apropiada es crucial para la clasificación precisa de géneros musicales. Se han desarrollado varias bases de datos para este propósito, cada una con sus ventajas y desventajas. Algunas bases de datos se centran en un conjunto específico de géneros musicales, mientras que otras cubren una amplia gama de géneros. Además, algunos conjuntos de datos están disponibles públicamente, mientras que otros requieren una licencia o pago.

En este estudio, se examinan varias bases de datos populares para la clasificación de géneros musicales, incluyendo la base de datos GTZAN, la base de datos de géneros musicales del Million Song Dataset y otros conjuntos de datos abiertos publicados en Kaggle. Se comparan estas bases de datos en términos de tamaño, diversidad de géneros, calidad de las etiquetas y accesibilidad. Además de la presencia de letras originales de las canciones.

Una vez elegida la base de datos, se procederá a su enriquecimiento con información adicional, como las características auditivas de las canciones que mejorarán la capacidad de la base de datos para la clasificación precisa de géneros y permitirá una mayor comprensión de sus factores.

El estudio presentado aborda un problema importante en el campo de la música y la informática: la clasificación de géneros musicales. Se examinan varias bases de datos populares y se elige una adecuada para su enriquecimiento posterior.

3.2 Revisión de Bases de Datos

Para llevar a cabo el estudio, se realizó una búsqueda exhaustiva de bases de datos disponibles en línea para su uso en la clasificación del género musical a través de las letras de las canciones. Durante la búsqueda, se identificaron varias bases de datos de canciones y música, incluyendo Million Song Dataset (MSD), Spotify Million Song Dataset (SMSD) y 5 Million Song Lyrics Dataset. Después de revisar estas opciones, se descartaron algunas debido a diversas limitaciones.

En primer lugar, se analizó la Spotify Million Song Dataset. Una base de datos de acceso abierto disponible en Kaggle, creada por el usuario Shrirang Mahajan. Sintetiza en cuatro columnas información completa sobre el dominio del problema, aportando en torno a cincuenta y ocho mil canciones con título, artista y letra de la canción. Sin embargo, este número tan limitado de registros sería uno de los motivos por los que se terminaría descartando.





▲ artist 	▲ song 	▲ link 	▲ text 
Artist's name	Song Name	Link to the song	Lyrics of the song
643 unique values	44824 unique values	57650 unique values	57494 unique values
ABBA	Ahe's My Kind Of Girl	/a/abba/ahes+my+kind+of+girl_20598417.html	Look at her face, it's a wonderful face And it means something special to me Look at the way t...
ABBA	Andante, Andante	/a/abba/andante+andante_20002708.html	Take it easy with me, please Touch me gently like a summer evening breeze Take your time, make...
ABBA	As Good As New	/a/abba/as+good+as+n	I'll never know why

Figura 3.1. Spotify Million Song Dataset

Uno de los principales inconvenientes de contar con un número tan pequeño de registros es la dificultad que surge al realizar la limpieza y el procesamiento de la base de datos. La limpieza de los datos es un paso esencial para eliminar errores, valores atípicos y registros incompletos o duplicados. Sin embargo, con una base de datos tan reducida, resulta complicado obtener conjuntos de entrenamiento lo suficientemente grandes y representativos.

Además, la cantidad limitada de registros en la base de datos dificulta la creación de modelos predictivos o de aprendizaje automático sólidos. Para obtener resultados precisos y generalizables, se requiere una cantidad significativa de datos de entrenamiento. Con menos de 58000 registros, es probable que el modelo resultante no alcance el nivel de rendimiento deseado debido a la falta de variabilidad y diversidad en los datos.

El otro principal motivo por el que finalmente se descartaría sería la falta de información acerca del género musical de las canciones.

Cuando se realiza una clasificación, es esencial utilizar características intrínsecamente vinculadas con la variable objetivo que se intenta predecir, en este caso, el género musical. Al carecer de información específica sobre el género en la base de datos, se dificulta enormemente la capacidad de realizar una clasificación precisa.

En el contexto de clasificar géneros musicales a partir de letras de canciones, se espera que la información sobre el género musical esté presente en la base de datos. Las letras pueden contener elementos distintivos, como palabras clave, referencias culturales o temáticas recurrentes, que pueden ser indicadores del género musical. Sin embargo, si una base de datos no proporciona esta información o no está etiquetada correctamente, la capacidad de realizar una clasificación precisa se ve comprometida.

Aunque se podría haber considerado la posibilidad de ampliar la base de datos, en vista de las circunstancias, se determinó que esta base de datos era limitada y no cumplía con los requisitos necesarios para nuestro estudio.

El siguiente conjunto de datos que se analizaría sería Million Song Dataset (MSD).

MSD es una recopilación exhaustiva de datos ampliamente utilizada en la investigación y el análisis en el ámbito de la música. Consiste en una colección de un millón de canciones obtenidas de diversas fuentes, junto con sus metadatos correspondientes. El propósito principal de este conjunto de datos es proporcionar una base de información sustancial para facilitar la investigación y el desarrollo de aplicaciones en el campo de la música, utilizando enfoques científicos y algoritmos avanzados.

Contiene información detallada sobre cada canción, incluyendo el artista, el álbum, el género musical, la duración, el año de lanzamiento y diversas características acústicas. Estas características acústicas se obtienen mediante el análisis digital de las señales de audio y abarcan aspectos como el tempo, la energía, la tonalidad y otros parámetros relevantes. Estos datos permiten un análisis profundo de las canciones y su estructura musical, así como el estudio de patrones y tendencias a gran escala en la música.

El uso de Million Song Dataset ha sido amplio y diverso en la comunidad científica y académica. Se ha utilizado para la investigación en áreas como la recuperación de información musical, la recomendación de música, la clasificación automática de géneros musicales, el análisis de popularidad y tendencias musicales, entre otros. Además, ha sido un recurso fundamental para el desarrollo de algoritmos y modelos en el campo del aprendizaje automático y la inteligencia artificial aplicada a la música. Es sin duda la opción más completa en términos de cantidad y calidad de información.

Sin embargo, su uso implica ciertas complejidades y requisitos adicionales en comparación con otras bases de datos de acceso abierto. Para utilizar el MSD, es necesario emplear un nodo de almacenamiento y recuperación de datos conocido como Amazon S3 (Simple Storage Service) para gestionar y almacenar los datos, los cuales ocupan aproximadamente 300 GB en total. Los investigadores y analistas deben adquirir conocimientos sobre los conceptos y herramientas relacionados con el almacenamiento y la indexación de datos en Amazon S3. Esto implica comprender los procedimientos para acceder, descargar y procesar los archivos del conjunto de datos, así como realizar operaciones de búsqueda y análisis específicas.

Por temas de optimización de tiempo se prefirió investigar bases de datos alternativas, evitando bloquear el desarrollo del proyecto en aprender de cero esta tecnología. Se mantendría la preferencia en plataformas como Kaggle que presentan mayor flexibilidad y facilidad de uso.

Finalmente, se optó por el 5 Million Song Lyrics Dataset (5MS), una base de datos de uso abierto en Kaggle creada por el usuario Nikhil Nayak.

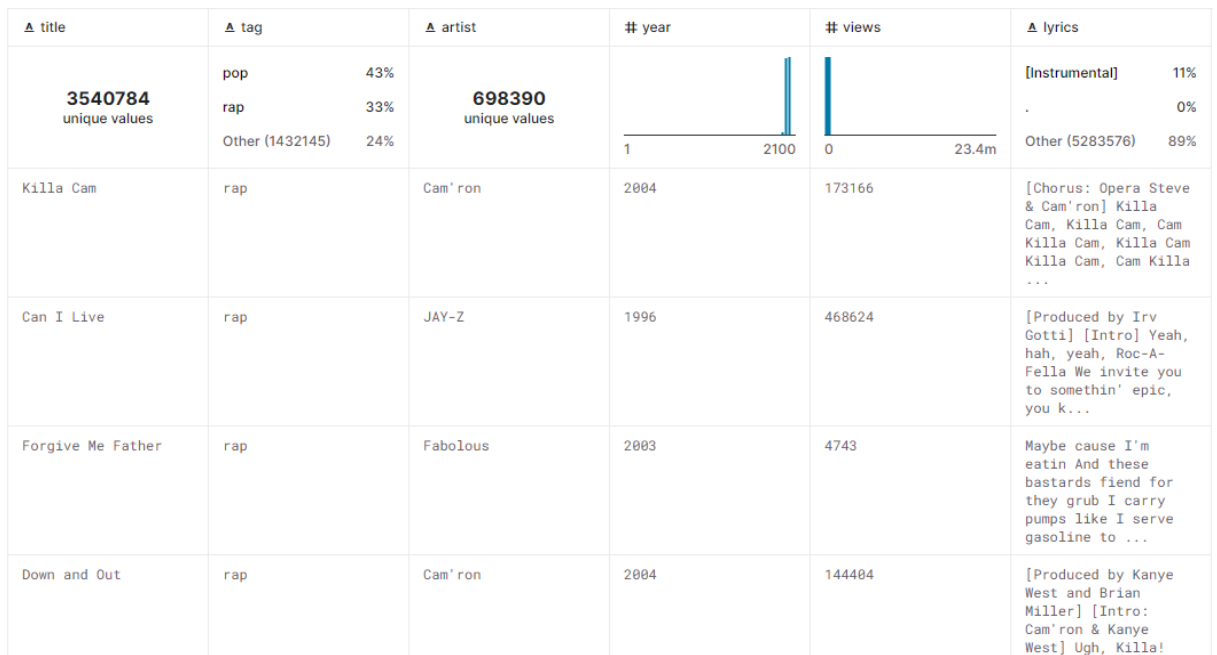


Figura 3.2. 5 Million Song Lyrics Dataset

Contenía una gran cantidad de registros, ofreciendo así una mayor flexibilidad para limpiar y balancear la base de datos, así como para enriquecerla posteriormente. Estos aspectos son fundamentales para obtener resultados fiables y significativos en el análisis de datos.

En primer lugar, la limpieza de datos es esencial para garantizar la calidad y la integridad de la información almacenada. Cuanto mayor sea el número de registros, más oportunidades existen para identificar y corregir errores, inconsistencias y valores atípicos en los datos. Esto incluye la eliminación de duplicados, la estandarización de formatos y la corrección de errores de entrada.

En segundo lugar, el balanceo de datos es importante para evitar sesgos y garantizar una representación equitativa de las distintas categorías o clases en un conjunto de datos. Un gran número de registros permite obtener una muestra más diversa y representativa de la población o fenómeno en estudio, lo que contribuye a obtener resultados más confiables y generalizables.

Por último, el enriquecimiento de los datos se refiere a la adición de información relevante y complementaria a los registros existentes. Contar con un gran número de registros en la base de datos original es de gran ayuda para poder aprovechar al máximo el enriquecimiento de datos mediante fuentes secundarias.

Cuando se busca enriquecer una base de datos utilizando fuentes externas, es necesario realizar comparaciones y coincidencias entre los registros de la base de datos original y los registros de la fuente secundaria. Al tener una base de datos original más grande, se aumenta la probabilidad de que existan registros coincidentes en la fuente secundaria, lo que permite obtener un enriquecimiento más completo y preciso.

Si la base de datos original es lo suficientemente grande, es más probable que contenga registros con información que no esté presente en la fuente secundaria. En este caso, sería posible descartar los registros que no estén informados y aun así mantener una cantidad significativa de datos válidos.

5 Million Song Dataset estaba compuesta por información detallada sobre las canciones, incluyendo su género, artista, año de lanzamiento y lo más importante, la letra de la misma obtenida desde las bases de datos de Genius.

Para utilizar el 5 Million Song Dataset, solamente se tuvo que añadir el Dataset Kaggle sobre nuestro kernel o libreta de preprocesamiento. En esta libreta se debería, en primer lugar, fijar el idioma de entrada de las letras de las canciones. 5MS contenía información de canciones en español, alemán, inglés, etc. por lo que era necesario filtrar por únicamente uno de ellos y centrar el estudio de las canciones en un único idioma.

Se creó un script haciendo uso de la librería SpaCy a partir del cual se analizaban las letras de las canciones y determinaba si estaban en inglés o no. Se decidió en última instancia filtrar por las canciones en inglés debido a que suponía un mayor número de registros en base de datos que el resto.

Para asegurarse de que los registros estuvieran completos y fuesen coherentes, se realizaron varias operaciones de limpieza y preprocesamiento. Se eliminaron los registros con valores faltantes, se balancearon las clases de género y se eliminaron las letras de las canciones que contenían errores o se consideraban irrelevantes para su clasificación.

Sería parte del posterior trabajo aumentar esta base de datos con información relativa a las características de audio. La inclusión de características de audio en la clasificación de géneros musicales a partir de las letras de las canciones desempeña un papel crucial en el análisis objetivo y preciso de la música. Estas características de audio se refieren a los atributos acústicos y estructurales de una grabación musical, como el timbre, la tonalidad, el ritmo, la energía y la dinámica, entre otros.

El uso de características de audio en combinación con las letras permite una comprensión más completa de una canción y facilita la identificación y clasificación de su género musical correspondiente. Las características de audio proporcionan información objetiva sobre los elementos sonoros y la expresión musical presentes en una canción, mientras que las letras aportan significado semántico y lírico.

El análisis de características de audio en la clasificación de géneros musicales es importante debido a varias razones. En primer lugar, las características de audio pueden revelar información sobre el estado emocional, el estilo interpretativo y las intenciones artísticas de una canción. Estos aspectos subjetivos de la música pueden complementar la información proporcionada por las letras y brindar una perspectiva más completa sobre la naturaleza y el propósito de una canción en particular.

Además, el uso de características de audio en la clasificación de géneros musicales permite un enfoque más objetivo y reproducible en comparación con el análisis basado exclusivamente en las letras. Las letras pueden ser ambiguas, subjetivas o contener referencias culturales que dificultan la clasificación precisa. Las características de audio, en cambio, se basan en mediciones objetivas y cuantificables, lo que facilita la aplicación de técnicas analíticas y algoritmos de clasificación.

Así conseguiríamos un conjunto de datos completo con el que poder realizar un estudio sólido y completo de la clasificación de géneros musicales.

3.3 Enriquecimiento de la Base de Datos

Una vez seleccionada la base de datos de 5 Million Song Dataset y preprocesada para eliminar registros incompletos o no relevantes para el análisis del género musical, se procedió a la tarea de aumentar la base de datos con información de características de audio de las canciones. Para ello, se hizo uso de Spotipy.

Spotipy es una biblioteca de Python que proporciona funcionalidades para interactuar con la API de Spotify. Una API es un conjunto de reglas y protocolos que permiten la comunicación entre diferentes sistemas informáticos. Actúa como una capa de abstracción que simplifica el acceso y manipulación de datos relacionados con la plataforma Spotify. Permite buscar canciones, obtener información sobre artistas, crear listas de reproducción y reproducir música a través de la API de Spotify.

Al utilizar Spotipy, se pueden aprovechar las capacidades de Spotify en diferentes aplicaciones y proyectos, sin necesidad de preocuparse por los detalles de implementación de la API subyacente. Spotipy se encarga de la comunicación con la API de Spotify, realiza solicitudes y procesa respuestas, facilitando la integración de la plataforma musical en el flujo de trabajo de desarrollo.

Se decidió utilizar Spotify como fuente complementaria a 5MS por la calidad de sus características de audio, entre las que se incluye tempo, energía, valencia, tono o instrumentación.

Se creó un script que, a través de Spotipy, obtenía las características de audio de cada canción de la base de datos. Primero buscaba a partir del título y artista si había registro de la canción en la plataforma o no. Si la búsqueda devolvía un resultado, recuperaba su identificador correspondiente y lo almacenaba en una lista. Si la búsqueda no devolvía ningún resultado, se marcaba la canción como no encontrada. Este proceso se repitió para cada canción en la base de datos.

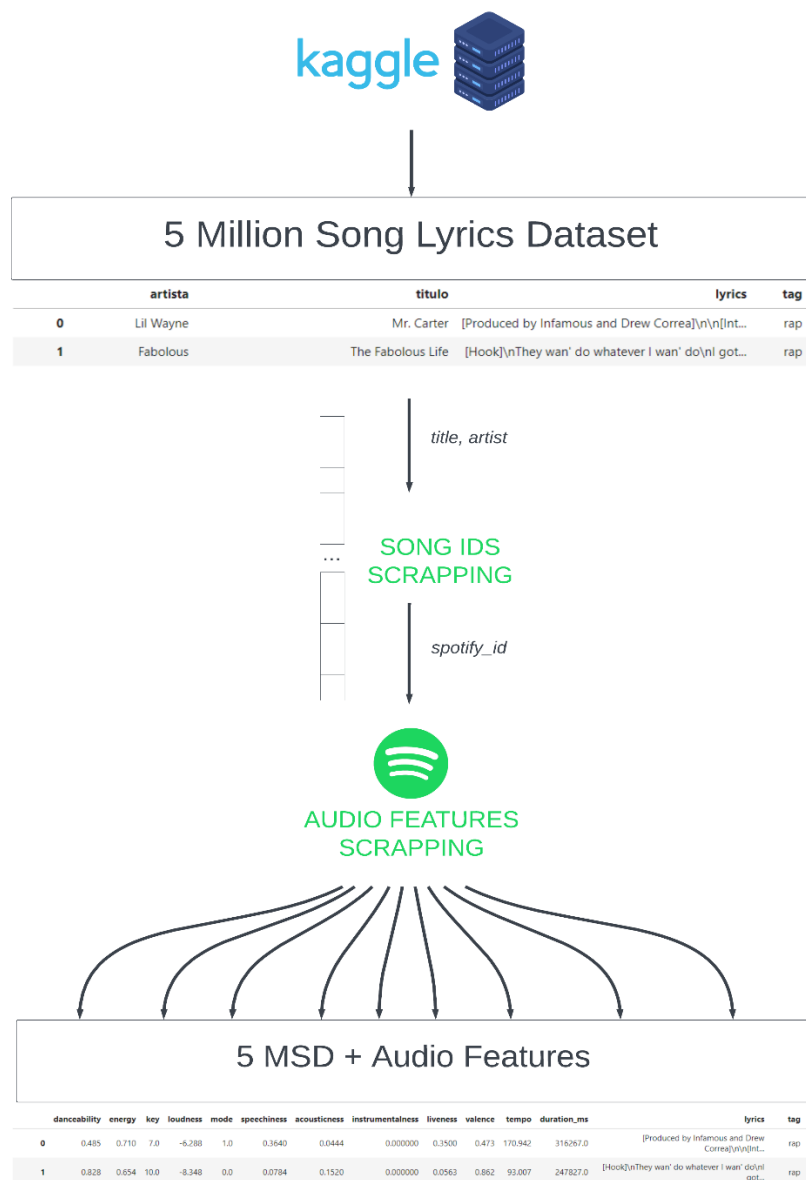


Figura 3.3. Flujo de extracción de características de audio

Dado que la API de Spotify tiene una limitación en la cantidad de llamadas que se pueden hacer en un período de tiempo determinado, se implementó un tiempo de espera entre cada llamada para evitar exceder los límites permitidos, previniendo así errores en la búsqueda.

Una vez obtenida la lista con los identificadores Spotify de 5MS solo bastaría extraer sus características de audio. Se hizo uso de la función *audio_features* de Spotipy que realiza consultas en lotes de hasta 100 identificadores y devuelve las características de audio correspondientes a cada una de sus canciones.

Una vez obtenida la información de las características de audio de cada canción, se agregaron estas características como nuevas columnas de 5MS.

	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	lyrics	tag	id
0	0.485	0.710	7.0	-6.288	1.0	0.3640	0.0444	0.000000	0.3500	0.473	170.942	316267.0	[Produced by Infamous and Drew Correa]\n\nIntro...	rap	2hQM18Urxp2kqOM6mN11TC
1	0.828	0.654	10.0	-8.348	0.0	0.0784	0.1520	0.000000	0.0563	0.862	93.007	247827.0	[Hook]\nThey wan' do whatever I wan' do\nI got...	rap	04kvMnJ6zo72bSnZDWyE7m
2	0.889	0.818	9.0	-4.639	1.0	0.2530	0.4700	0.000000	0.1790	0.782	96.063	302760.0	[Intro: The Notorious B.I.G.]\n("Fuck all you ...	rap	5ByAJIEEnvYdvnezg7HTX
3	0.612	0.850	1.0	-4.703	1.0	0.5000	0.0393	0.000000	0.2550	0.781	173.497	204027.0	[Chuck D]\n1, 2, 3, 4, 5, 6, 7, 8, 9\n\nIntro...	rap	1Z7C8CIE8UEaH70jCceJH2
4	0.792	0.694	8.0	-8.496	0.0	0.3300	0.0619	0.000023	0.2500	0.804	177.094	309773.0	[Intro: JAY-Z]\nWhat?nWell fuck you... bitch\...	rap	4LGMSdeKOUoy5W75Je0HI
...
302213	0.417	0.327	7.0	-12.678	1.0	0.0321	0.6530	0.608000	0.2290	0.394	84.681	167400.0	[Instrumental]	rb	29EAgZQvXMqwm2O81MffjV
302214	0.309	0.871	5.0	-3.556	1.0	0.0867	0.0972	0.000000	0.1970	0.550	84.574	191478.0	[Verse 1: Julia Cole, Cooper Alan]\nHate hangi...	country	46xwJOFMgB1biZv3rztI4pT
302215	0.632	0.188	11.0	-13.764	0.0	0.0364	0.6640	0.002970	0.4810	0.573	88.876	123200.0	[Instrumental]	rb	6ytlLFQv1NVafUWSkmE2S
302216	0.352	0.790	2.0	-2.915	1.0	0.0319	0.0719	0.000000	0.0714	0.320	74.973	200000.0	[Verse 1]\nHalf truth and half you\nDidn't we ...	pop	2bVh1aGyCWaKkZvEn61xZ1
302217	0.407	0.523	0.0	-6.431	1.0	0.0331	0.4540	0.000000	0.0874	0.336	138.958	166427.0	[Verse 1]\nYou need a new number, one that ain...	country	6xboopPn3bSj93xZ9N68ZM

Figura 3.4. 5MS tras el enriquecimiento con características de audio

Este proceso de adición de características de audio permitió enriquecer la información disponible de cada canción, lo que mejoró la capacidad del modelo de clasificación para discernir entre géneros similares. Además, al contar con esta información, se abrieron nuevas posibilidades para la exploración de los datos y la realización de análisis más detallados.

Capítulo 4

Experimentación

4.1 Análisis Exploratorio de Datos

Antes de abordar la construcción y entrenamiento de los modelos para el problema en cuestión, se llevó a cabo un análisis exploratorio de la base de datos final. Durante este proceso se emplearon librerías Python, tales como Pandas o Seaborn, entre otras.

De acuerdo con el siguiente gráfico, se pudo observar que la base de datos presentaba una distribución desbalanceada de registros por clase. Es decir, no existía el mismo número de canciones por género en todos los casos. A pesar de ser una de las principales preocupaciones consideradas durante la fase de preprocesamiento, se encontró que no todos los géneros se comportaban de la misma forma durante este proceso. Por ejemplo, la clase misc o "Miscellaneous" contaba con muchos menos registros que las demás, posiblemente debido a la falta de información dentro de las bases de datos de Spotify.

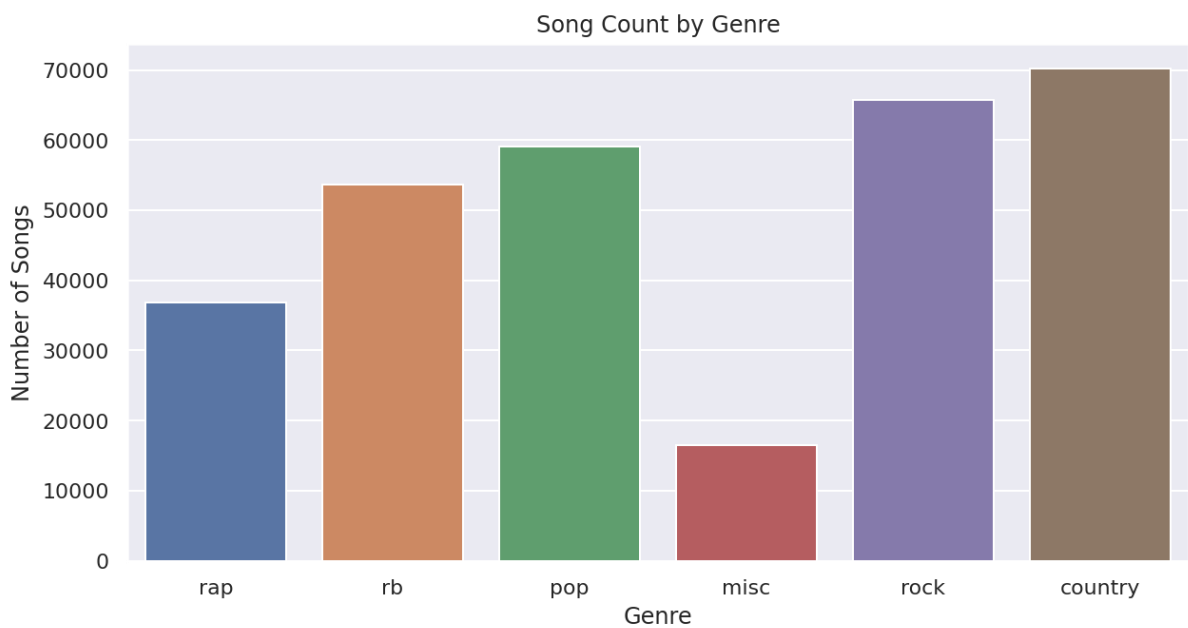


Figura 4.1. Distribución de géneros musicales

Otro de los gráficos más comunes en el análisis exploratorio de características de audio suele ser el de dispersión entre las variables valencia y energía.

Según la teoría de la emoción en la música, ésta puede transmitir emociones y sentimientos a los oyentes (Scherer, 1993). La valencia es una medida que refleja la positividad musical transmitida por una canción, y se mide en una escala de 0.0 a 1.0. Las canciones con una alta valencia suelen sonar más positivas, mientras que las canciones con una baja valencia tienden a sonar más negativas (Juslin y Västfjäll, 2008).

Por otro lado, la energía en la música se refiere a la cantidad de actividad o intensidad que una canción puede tener. Algunas canciones tienen una alta energía, lo que significa que son rápidas y con ritmo, mientras que otras tienen una baja energía, lo que significa que son más lentas y tranquilas (North, Hargreaves y O'Neill, 2000). La energía también puede influir en las emociones que se despiertan en los oyentes, ya que las canciones más enérgicas tienden a evocar emociones más fuertes (Herrera, Shneiderman y Preece, 2010).

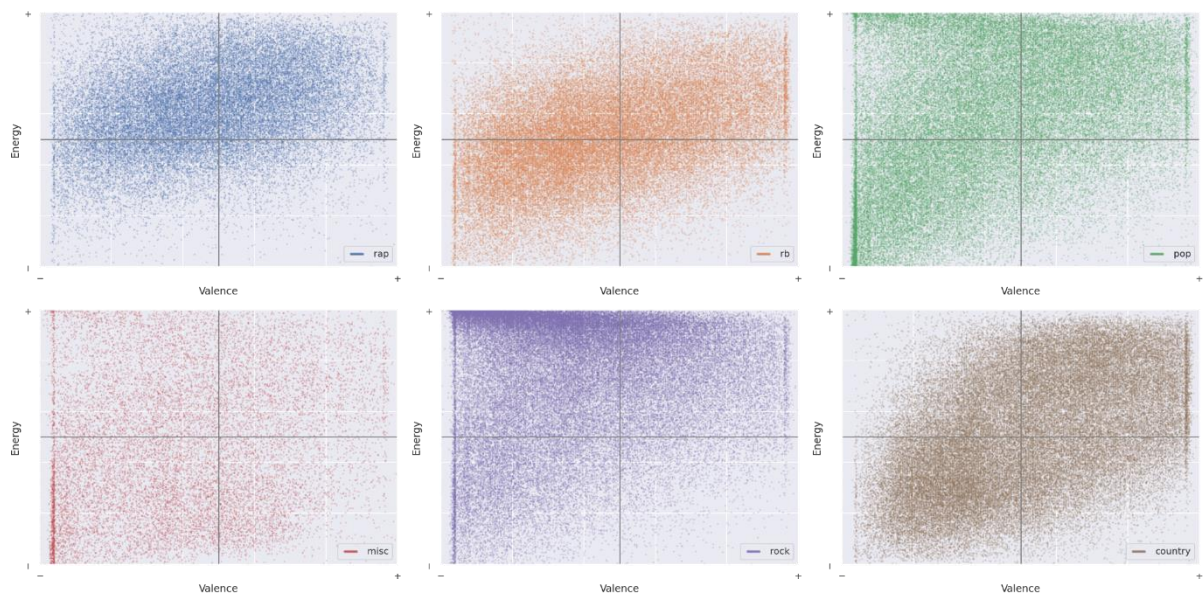


Figura 4.2. Gráfico de dispersión Energía / Valencia

Se puede observar como quizá las canciones rock (gráfico de color morado) son más negativas, pero más enérgicas y rápidas. Mientras que en el resto de géneros no se aprecian claramente distribuciones discriminatorias o singulares.

Al analizar individualmente la distribución de la energía entre los distintos géneros musicales, se observa que rap, pop y rock cuentan con los valores más altos. Concretamente rock superando el 0.6 de energía.

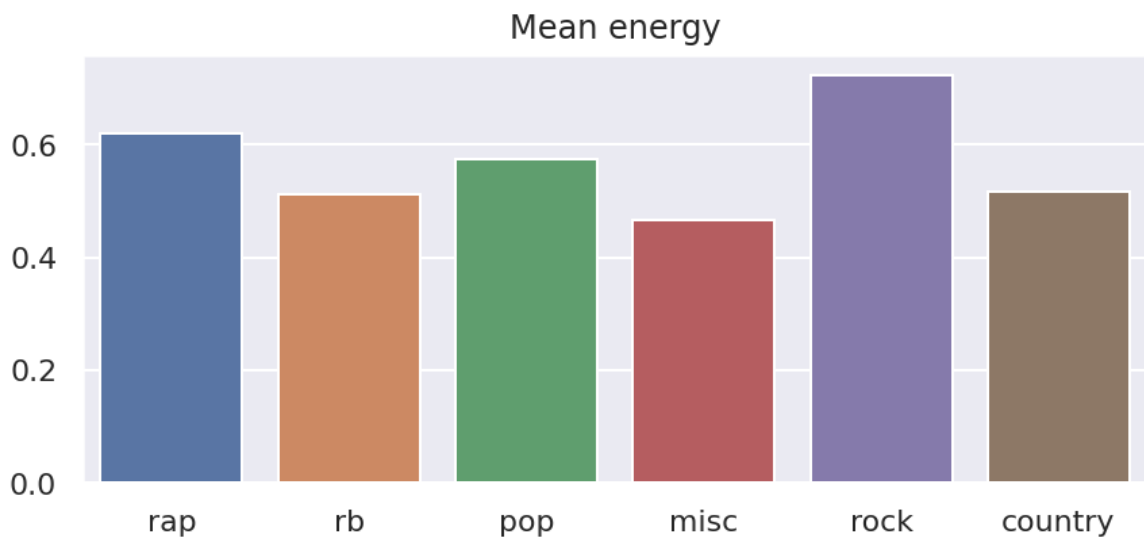


Figura 4.3. Distribución de Energía por Género

Generalmente el género musical rock exhibe un índice de energía superior en comparación con otros géneros como el rap o el pop debido a diversas características musicales y culturales que lo distinguen.

En primer lugar, desde una perspectiva musical, el rock se caracteriza por su ritmo enérgico y contundente, que se fundamenta en la utilización de instrumentos como la guitarra eléctrica, la batería y el bajo. Estos instrumentos, en combinación con técnicas de ejecución como riffs poderosos, solos de guitarra virtuosos y secciones rítmicas intensas, generan una sensación de fuerza y potencia en la música rock. Además, el uso de amplificación y efectos distorsionados en la guitarra contribuye a aumentar la sensación de energía.

Por otro lado, el contenido lírico del rock a menudo aborda temas emocionales y rebeldes, expresando sentimientos de frustración, pasión, deseo y descontento social. Esta temática emocionalmente cargada y en ocasiones provocativa puede generar una respuesta emocional intensa en el oyente, contribuyendo a la percepción de mayor energía en el género.

Desde una perspectiva cultural, el rock ha evolucionado históricamente como una forma de expresión musical asociada a actitudes rebeldes y contraculturales. La subcultura del rock se ha relacionado con valores de libertad, individualismo y resistencia, lo cual puede influir en la percepción de una energía intrínseca al género.

Otra de las características auditivas que más información aporta a los datos es 'speechiness'. Representa dentro del umbral 0.0 y 1.0 la presencia de palabras dentro de una la pista de una canción. Cuantas más palabras se detecten en la canción mayor será este valor.

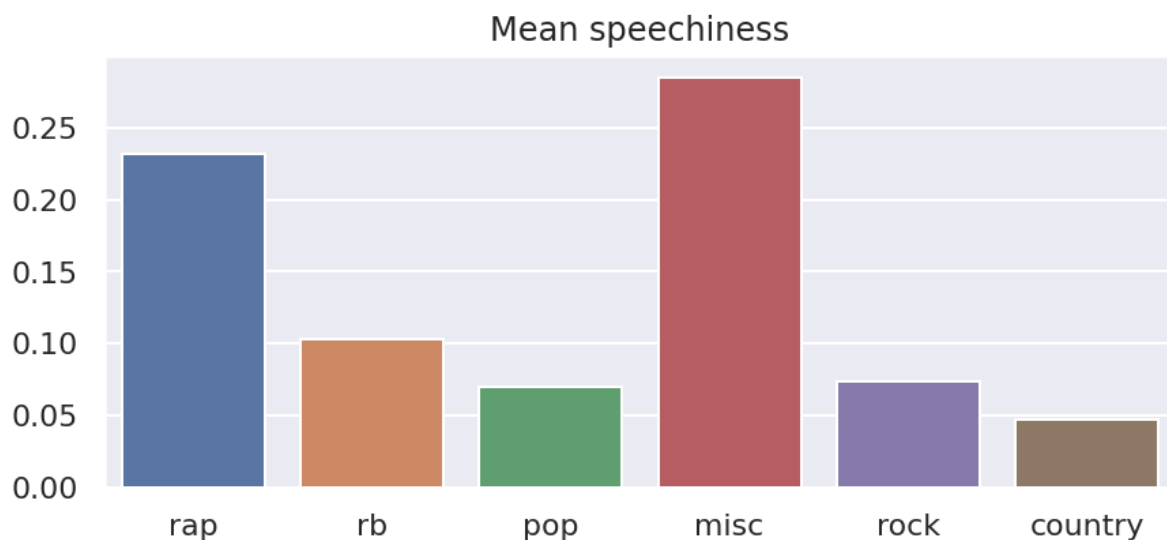


Figura 4.4. Distribución de Speechiness por Género

Al mostrar la distribución por la media de 'speechiness' en cada género se aprecia como algunos géneros como pop, rock o country cuentan con un índice de speechiness mucho menor que rap, por ejemplo. Y es que el rap tiende a exhibir un índice de "speechiness" o hablabilidad más alto debido a una serie de factores relacionados tanto con sus características estilísticas como con su contexto histórico y cultural.

En primer lugar, el rap se caracteriza por su enfoque en la palabra hablada y la expresión lírica. Los artistas de rap suelen utilizar patrones rítmicos y rimas para transmitir mensajes y contar historias de una manera altamente estructurada. Esto implica que el énfasis se coloque en la vocalización clara y articulada de las letras, lo que se traduce en un mayor índice de "speechiness".

Además, el rap tiene sus raíces en las tradiciones de la poesía y la oratoria, donde el dominio del lenguaje y la habilidad verbal son altamente valorados. Los artistas de rap suelen ser elocuentes y utilizan técnicas como la métrica, la aliteración, la asonancia y la improvisación para realzar su habilidad de expresión verbal. Esto se traduce en una mayor cantidad de palabras pronunciadas por minuto y, por lo tanto, en un índice de "speechiness" más elevado.

Otro aspecto importante es el contexto cultural del rap. Históricamente, el rap ha sido una forma de expresión para las comunidades marginadas y desfavorecidas, que utilizan la música como una herramienta para transmitir mensajes sociales, políticos y personales. Dado que las letras y la entrega vocal son componentes fundamentales en la comunicación de estos mensajes, los artistas de rap han desarrollado una habilidad excepcional para articular sus pensamientos de manera clara y directa, aumentando así el índice de "speechiness".

Por otro lado, el rock y el country, aunque también utilizan la voz como elemento fundamental en su música, se enfocan más en la melodía y la instrumentación. Estos géneros suelen tener estructuras de canciones más largas y se centran en la expresión musical a través de los instrumentos. Esto puede dar lugar a una menor proporción de palabras por minuto y, por lo tanto, a un índice de "speechiness" más bajo en comparación con el rap.

Al observar la gráfica también se puede apreciar como miscellaneous presenta un índice de speechiness alto. Miscellaneous es la etiqueta utilizada en nuestra base de datos para encapsular las canciones que no se pueden identificar con uno de los otros cinco géneros. Abarca una amplia gama de estilos y subgéneros musicales que no se adscriben claramente a categorías predefinidas. Algunos de estos, como el cabaret, el teatro musical y la ópera, están intrínsecamente basados en el despliegue de actuaciones vocales y habilidades interpretativas, lo que conduce a una mayor presencia de elementos vocales en sus grabaciones.

Estos subgéneros se caracterizan por su énfasis en las habilidades vocales, donde la narrativa y la emotividad expresada a través de la voz son elementos esenciales para transmitir la intención artística y contar historias.

Este es el caso, por ejemplo, de los dos registros que se muestran a continuación:

	speechiness	tempo	duration_ms	tag	id
129	0.913	101.445	509680.0	misc	2cPLZlw3CPvvu8Kn13x18p
3379	0.926	134.707	1822772.0	misc	3pCg6GICGcZQGFnJWbWWJT

Figura 4.5. Casos curiosos Speechiness

“The Waste Land” de T.S. Eliot y “Act One – Julius Caesar” de William Shakespeare, respectivamente. Composiciones literarias en prosa y verso que emplean la voz de un narrador para transmitir la trama y los elementos de la historia.

Con respecto a las letras de las canciones, se hizo uso de wordclouds. Una "wordcloud" (también conocida como nube de palabras o nube de etiquetas) es una representación visual de las palabras más comunes en un texto o conjunto de textos. En una wordcloud, el tamaño de cada palabra se determina por la frecuencia de su aparición en el texto. Las palabras más comunes aparecen en un tamaño mayor y a menudo en negrita, mientras que las palabras menos comunes aparecen en un tamaño más pequeño.

Se usó la librería wordcloud de Python a la que se le pasarían, a través de su función WordCloud, las canciones ya limpias de cada género y generaría las imágenes correspondientes a sus wordclouds.



Figura 4.6. Wordclouds por género

Finalmente, se procedería a la división de la base de datos en los conjuntos de entrenamiento y prueba. El conjunto de entrenamiento se utilizaría para que el modelo pueda aprender y ajustarse a través de la exposición a casos conocidos. Por otro lado, el conjunto de prueba se emplearía como un conjunto de producción, que consiste en casos nuevos y desconocidos para el modelo, y se utiliza para evaluar su desempeño en situaciones reales. Es esencial que ambos conjuntos sean establecidos de manera precisa y fija, con el fin de evitar problemas de fuga de datos o comportamientos engañosos por parte del modelo.

```
Training data samples: 241774
Test data samples: 60444
```

Figura 4.7. Distribución final Train y Test

4.2 Machine Learning

Se construyeron dos modelos de clasificación utilizando las características de audio para deducir el sistema de reglas final para la clasificación de géneros musicales. Se utilizó la librería scikit-learn de Python para implementar ambos modelos.

Para imputar y estandarizar los datos de entrada, se construyó una Pipeline que incluiría la imputación de valores faltantes y la escalada de los datos. En problemas de clasificación, la estandarización de las variables de entrada es importante ya que reduce la influencia de las variables que presentan rangos de valores muy diferentes.

Además, para realizar el proceso de búsqueda exhaustiva de los mejores hiperparámetros, se hizo uso de un objeto GridSearchCV de scikit-learn al que se le pasarían los modelos a entrenar junto con sus hiperparámetros.

Una vez creada la Pipeline y el buscador de los mejores hiperparámetros se crearon y entrenaron los dos modelos.

El primero sería un árbol de decisión. El árbol de decisión se basa en la construcción de un árbol binario, donde cada nodo representa una característica o variable y cada rama representa una decisión o resultado basado en esa característica.

En un árbol de decisión, se comienza con un nodo raíz que contiene el conjunto completo de datos. Luego, se selecciona la característica más importante para dividir los datos en dos subconjuntos, de tal manera que los subconjuntos resultantes sean lo más homogéneos posible. Este proceso se repite recursivamente para cada subconjunto hasta que se alcanza un criterio de parada, como un número mínimo de muestras en un nodo o la profundidad máxima del árbol.

Una vez construido el árbol, se puede utilizar para hacer predicciones en nuevos datos. Para hacerlo, se comienza en la raíz del árbol y se sigue el camino correspondiente a cada característica en los datos de entrada, hasta llegar a una hoja que contiene la predicción final.

A través de la clase `DecisionTreeClassifier` de `scikit-learn` se creó el modelo base de árbol de decisión que tras un primer entrenamiento no devolvió más de un **40.53%** de accuracy. Se definirían entonces los hiperparámetros a ajustar:

- **`imputer_strategy = ['mean', 'median']`**: Si el imputador de valores faltantes utilizaría la media o la mediana de los datos para completar.
- **`max_depth = [None, 3, 5, 7, 9]`**: La profundidad máxima que puede alcanzar el árbol durante las ramificaciones del entrenamiento.
- **`class_weight = [None, 'balanced']`**: Si el algoritmo debe ajustar automáticamente los pesos de las decisiones en torno a la frecuencia de la clase o no.

Tras recorrer una 5 validación cruzada dentro del `GridSearchCV` se obtuvo que los mejores hiperparámetros eran **'mean', 9 y None** respectivamente. Tras el segundo entrenamiento, ya con la mejor combinación de hiperparámetros aplicada, el árbol de decisión mejoró más de un 10% en accuracy, obteniendo exactamente un **50.94%**. Lo que demuestra sin duda la importancia de la búsqueda exhaustiva de hiperparámetros en fases de preentrenamiento.

Para el conjunto de test el Decision Tree obtuvo en torno al **51.10%**. Demostrando robustez frente a los nuevos casos.

El experimento al completo de entrenamiento y test del modelo llevó no más de **10 minutos** en su ejecución. Este será otro de los factores determinantes a comparar frente al resto de modelos del estudio.

El segundo modelo fue un random forest. El modelo random forest es una extensión del árbol de decisión que utiliza múltiples árboles para mejorar la precisión y reducir la variabilidad. Cada árbol se entrena con una muestra aleatoria de los datos de entrada y utiliza un subconjunto aleatorio de características para tomar decisiones en cada nodo. Al final, las predicciones se combinan mediante votación o promediado para obtener una predicción final.

Una ventaja clave del modelo random forest es su capacidad para manejar datos faltantes y variables categóricas. Además, al utilizar múltiples árboles, el modelo puede capturar relaciones no lineales entre las características y prevenir el sobreajuste.

En scikit-learn, se puede crear un modelo random forest utilizando la clase `RandomForestClassifier`. Al igual que con el árbol de decisión, hay varios hiperparámetros que se pueden ajustar para optimizar el rendimiento del modelo.

Uno de los particulares de este caso es el **n_estimators**, que fija el número de árboles que contendrá el bosque en su interior. Además de **n_estimators** también se buscará optimizar la máxima profundidad de cada árbol individualmente. Los otros dos hiperparámetros, **imputer_strategy** y **class_weight** se mantienen fijos según el ajuste de hiperparámetros visto en el modelo anterior.

- **imputer_strategy = 'mean'**
- **n_estimators = [100, 300]**
- **max_depth = [None, 3, 5, 7, 9]**
- **class_weight = None**

Concluyendo que con un número de árboles en su interior de **300** y sin restringir la profundidad máxima de los mismos se optimizaba el accuracy en entrenamiento, **54.94%**. Parece que el modelo se comporta más adecuadamente con un mayor número de modelos en su interior y sin una profundidad fija en cada uno de ellos.

Al pasarle el conjunto de test, el Random Forest clasifica correctamente un **55.13%** de los casos. Esto indica, como en el modelo anterior, que se ha entrenado un modelo robusto a casos desconocidos.

Se comparan ahora ambos modelos:

Tabla 4.1. Resultados Machine Learning

	ACC. PRE-AJUSTE	ACC. POST-AJUSTE	ACC. TEST	TIEMPO
Decision Tree	40.53%	50.94%	51.10%	9min 36s
Random Forest	53.41%	54.94%	55.13%	2h 26min 6s

Se puede apreciar que ambos modelos mejoran adecuadamente sus métricas de entrenamiento tras el ajuste de hiperparámetros. Además, también se puede apreciar que ambos se comportan adecuadamente frente a casos desconocidos.

Sin embargo, hay una clara diferencia en términos de tiempo de entrenamiento de cada modelo. Tardando el Decision Tree un **93.42%** menos en entrenar que el Random Forest. Esto se debe a que el segundo es una extensión escalada del primero.

4.3 Deep Learning

Se construyeron dos modelos de aprendizaje profundo utilizando las letras de las canciones para entrenar un modelo de procesamiento natural del lenguaje. Se utilizó la biblioteca Keras de Python para implementar ambos modelos.

Se creó una función de preprocesamiento con la que se limpiarían las letras de las canciones de información irrelevante en su interior como signos de puntuación, anotaciones en el texto o palabras redundantes. Se hizo uso de librerías como `re` (Regular Expressions) o `nltk` (Natural Language Toolkit), entre otras. Además de funciones nativas python como `map` y la de la clase `String`, `translate`.

Al aplicarla se obtendrían las letras de las canciones en formato vector de palabras con solamente las palabras que aportasen información al problema. Esto fue de ayuda para crear posteriormente la capa de embedding que compartirían ambos modelos.

Esta capa de embedding devuelve vectores numéricos densos con las relaciones semánticas entre palabras. Consiguiendo una estructura de información compacta y muy completa sobre las lyrics de las canciones. La salida de esta capa irá conectada a la arquitectura particular de cada uno de los modelos.

El primer modelo fue una red neuronal convolucional (CNN). Más concretamente una red neural convolucional unidimensional (CNN - 1D). A diferencia de las redes neuronales convolucionales clásicas que están diseñadas para procesar datos con estructura de cuadrícula, como imágenes, éstas se utilizan para analizar datos unidimensionales, como series de tiempo o secuencias de textos. Las redes convolucionales unidimensionales son especialmente efectivas para capturar patrones locales y estructuras secuenciales en los datos.

La idea principal detrás de las redes convolucionales 1D es la convolución, que es una operación matemática fundamental. La convolución consiste en aplicar un filtro o kernel a través de la secuencia de datos de entrada para extraer características relevantes. El filtro es una ventana deslizante que se va moviendo a lo largo de la secuencia, multiplicando sus valores por los valores correspondientes de los datos de entrada y sumando los resultados.

Al aplicar múltiples filtros a los datos de entrada, se generan múltiples "mapas de características" que capturan diferentes aspectos o patrones de la secuencia. Estos mapas de características se combinan mediante operaciones como la agrupación o pooling para reducir la dimensionalidad y preservar las características más importantes.

Después de la etapa de extracción de características, se pueden agregar capas adicionales, como capas totalmente conectadas, para realizar la clasificación o realizar otras tareas específicas.

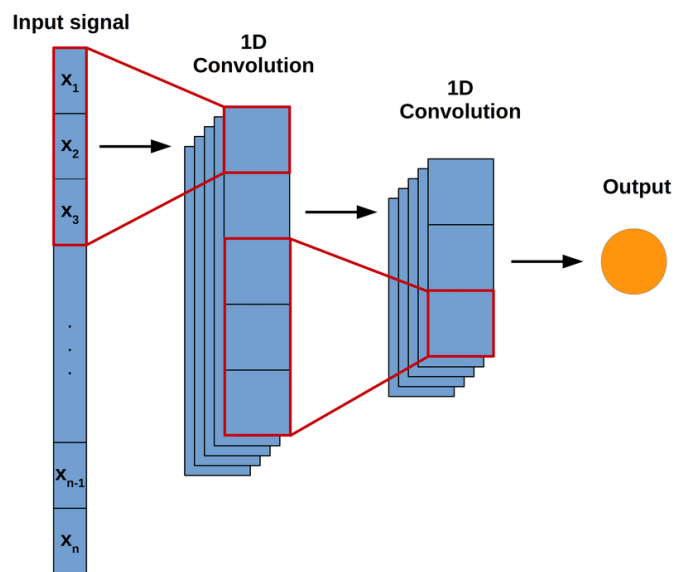


Figura 4.8. Diagrama Redes Convolucionales Unidimensionales - 2

Tras varios experimentos con diferentes estructuras de capas, el modelo quedó como sigue:

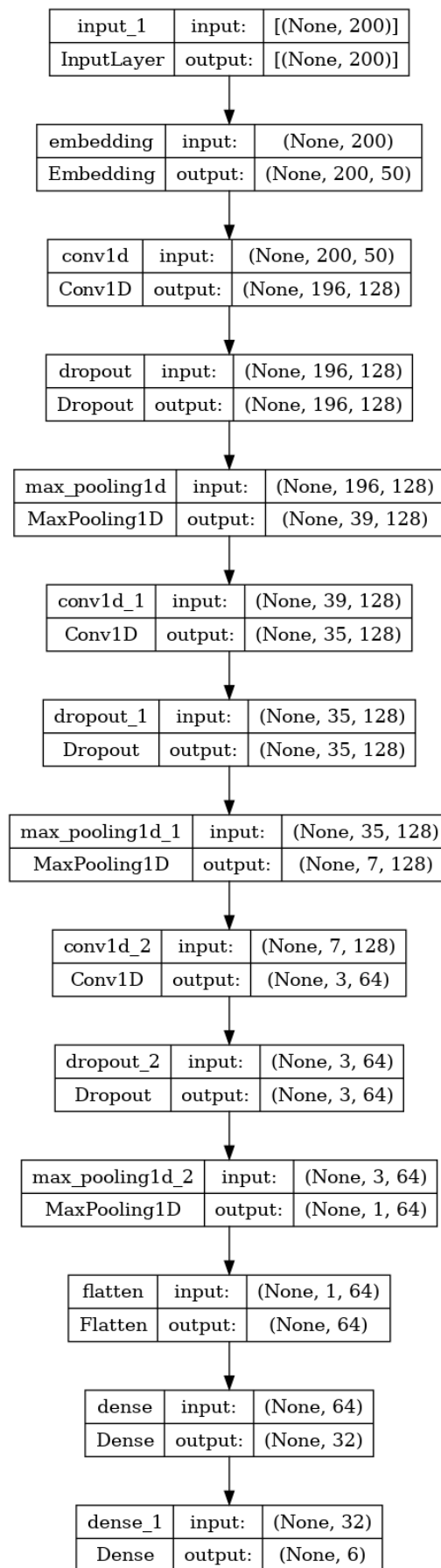


Figura 4.9. Arquitectura Keras Red Convolucional

Se aplican 3 capas convolucionales unidimensionales seguidas cada una de ellas por capas de agregación max-pooling, además de capas de dropout intercaladamente. Las capas de Dropout son una técnica de regularización que desactiva aleatoriamente neuronas durante el entrenamiento para evitar el sobreajuste. Ayudan a obtener redes neuronales más robustas y generalizables al forzar una distribución equilibrada del aprendizaje entre las neuronas.

A continuación, se aplica una capa Flatten para transformar el tensor multidimensional en un vector unidimensional. La capa Flatten "aplana" la salida de la CNN 1D, manteniendo la información y conservando la estructura espacial de las características aprendidas.

Después de aplicar la capa Flatten, los datos se pueden conectar directamente a las capas densas. Comúnmente se suelen encontrar una o varias capas densas previas a la capa final del modelo.

En este caso se aplica una de 32 neuronas que realiza la combinación de características extraídas por las capas convolucionales anteriores. Con ella se extraen relaciones más complejas, no lineales de los datos que se conectan directamente con la capa final de salida de predicciones. El número de neuronas de esta última capa viene fijo por las etiquetas a clasificar, los 6 géneros musicales.

Se hizo uso de un Learning Rate adaptativo a través de la función callback de keras ReduceLROnPlateau para optimizar el entrenamiento.

El learning rate (tasa de aprendizaje) es un hiperparametro crítico que determina la magnitud del ajuste de los pesos en cada iteración durante el entrenamiento de un modelo. Un learning rate demasiado pequeño puede hacer que el entrenamiento sea lento y tarde mucho tiempo en converger, mientras que un learning rate demasiado grande puede causar que el modelo salte alrededor del mínimo óptimo y no logre converger.

En lugar de utilizar un learning rate fijo para todo el proceso de entrenamiento, los métodos de learning rate adaptativo ajustan automáticamente el learning rate a medida que se avanza en el proceso de entrenamiento. La intuición detrás de esto es aprovechar la información del proceso de optimización para ajustar el learning rate de manera más adecuada.

```
reduce_lr = keras.callbacks.ReduceLROnPlateau(monitor='val_accuracy',  
                                              factor=0.2,  
                                              patience=2,  
                                              min_lr=0.000001)
```

Figura 4.10. Código Callback Learning Rate Adaptativo

Se puede observar en la siguiente gráfica la evolución de la precisión a lo largo del entrenamiento como desde épocas bastante tempranas el modelo consigue converger sobre el accuracy de entrenamiento. Una gráfica de precisión que demuestra un modelo prácticamente sin sobreajuste.

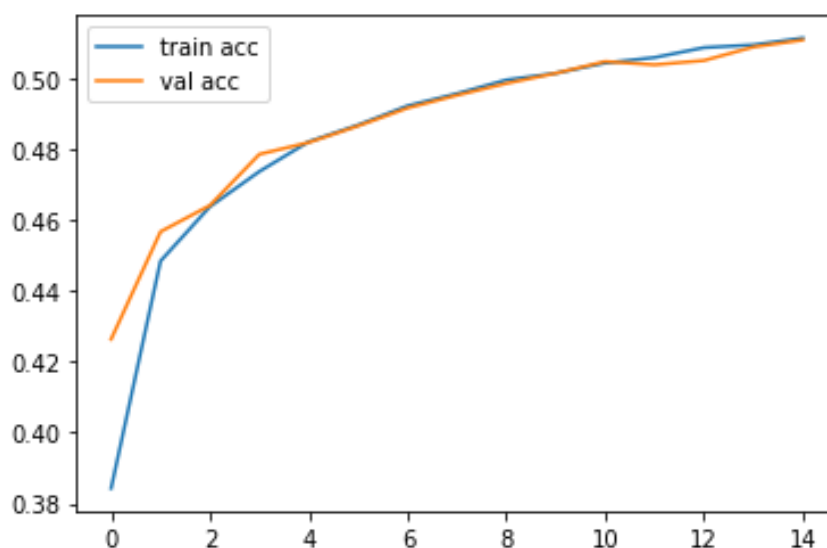


Figura 4.11. Gráfico de evolución Red Convolutiva

Finalmente, y tras 15 épocas de entrenamiento, se obtuvo un accuracy de en torno al **51.78%**.

Se probó entonces el modelo CNN unidimensional sobre el conjunto de test. Obteniendo un porcentaje de acierto frente a canciones nuevas del **52.41%**.

El segundo modelo es una red neuronal de memoria a largo plazo (LSTM). La LSTM se construyó utilizando una arquitectura con varias capas LSTM y capas densas.

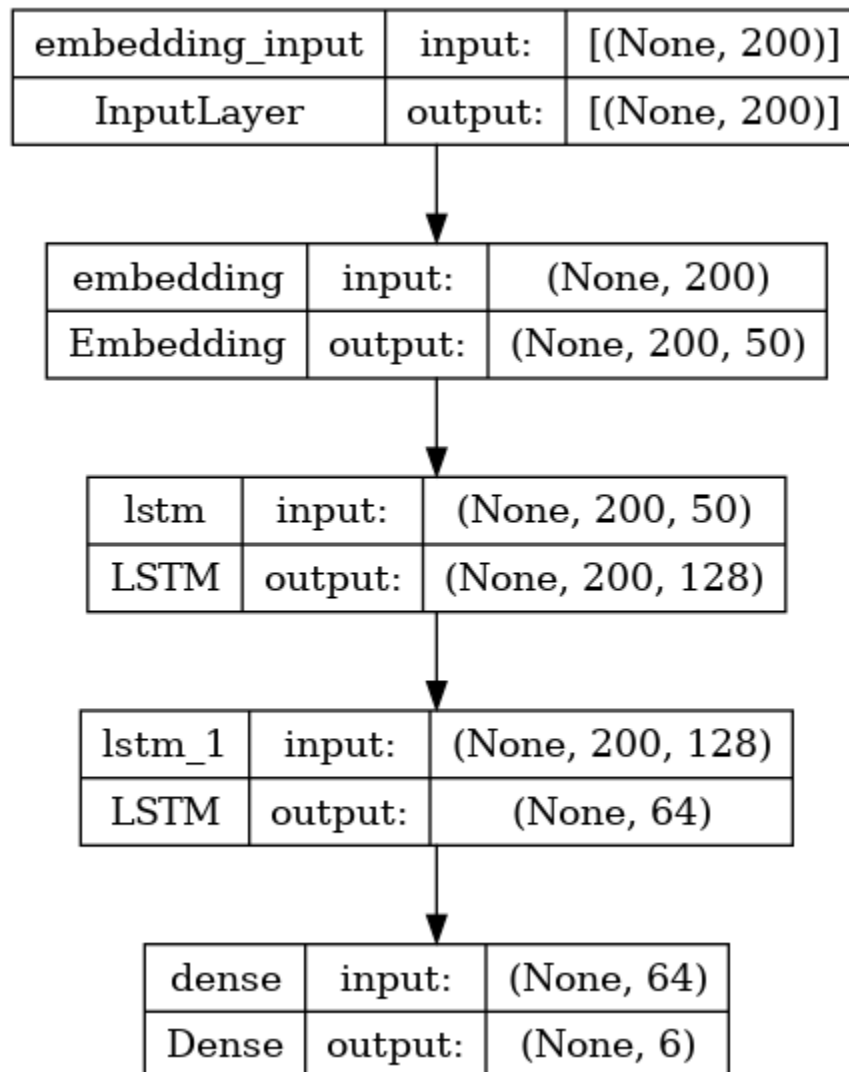


Figura 4.12. Arquitectura Keras Red de Memoria a Corto Plazo

Se llevaron a cabo pruebas utilizando diversas combinaciones de capas, así como diferentes valores de dropout y learning rates, con el objetivo de encontrar la estructura óptima para el modelo.

En este caso, el entrenamiento de las LSTM resulta mucho más lento computacionalmente que el de las CNN 1D. Esta diferencia en velocidad puede atribuirse a la arquitectura inherente de los modelos. Las CNN 1D se caracterizan por su capacidad de paralelizar el procesamiento mediante capas convolucionales y de pooling, lo que permite un entrenamiento más eficiente en términos computacionales. En contraste, las redes LSTM son estructuras recurrentes que procesan los datos de forma secuencial, lo que implica una mayor complejidad computacional y tiempos de entrenamiento más prolongados.

Por ello y por motivos de optimización de tiempos del estudio se decidió fijar un entrenamiento de 10 épocas en este caso.

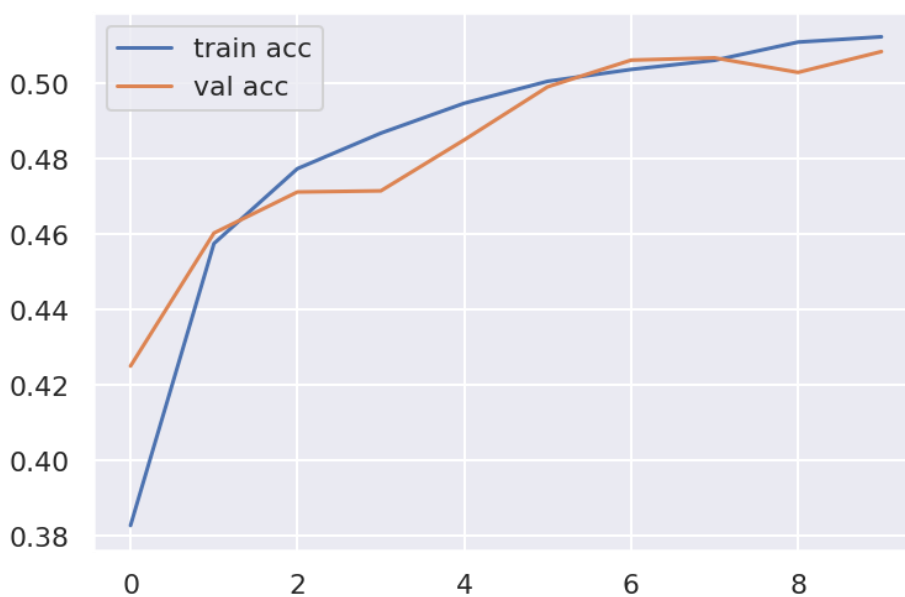


Figura 4.13. Gráfico de evolución Red de Memoria a Corto Plazo

El comportamiento en este caso es más irregular que en el anterior. El accuracy en validación no converge con el de entrenamiento como pasaba antes, en épocas tempranas y aumentando de forma prácticamente paralela. Quizá sería interesante valorar una parada temprana del entrenamiento sobre la época 5.

Tras el entrenamiento completo del modelo, el LSTM logró un accuracy del **50.52%**.

Comparando esta métrica sobre la del conjunto de test se observa una vez más un modelo sólido capaz de clasificar correctamente un porcentaje similar al de entrenamiento de canciones completamente nuevas, **51.32%**.

Tabla 4.2. Resultados Deep Learning

	ACC. TRAIN	ACC. TEST	TIEMPO
Decision Tree	50.94%	51.10%	9min 36s
Random Forest	54.94%	55.13%	2h 26min 6s
CNN 1D	51.78%	52.41%	2h 17min 23s
LSTM	50.52%	51.32%	8h 38min 15s

Después de entrenar y evaluar los modelos correspondientes a cada parte se concluye que Random Forest es el mejor modelo para clasificar géneros musicales a partir de las características de audio de las canciones. Sin embargo, en términos de tiempo de entrenamiento el Decision Tree hace un trabajo mucho más eficiente que su versión escalada.

Por otro lado, para la parte de las letras de las canciones el modelo NLP que obtiene mejor accuracy en este problema es el CNN unidimensional. Además de ser considerablemente más rápido que el de red recurrente LSTM, concretamente un **73.49%**.

4.4 Ensemble

El modelo final que se presentará en este estudio será una combinación de los modelos de las dos partes. En principio, se supondrá que la mejor opción será la que optimiza en ambas partes la métrica de precisión. Es decir, Random Forest para las características de audio y CNN 1D para las letras de las canciones. Igualmente se estudiarán el resto de combinaciones y evaluarán comparativamente.

Se seguirán tres formas de combinar ambos modelos: por la media aritmética de los resultados en test de cada uno de ellos, a través de un tercer modelo que combine dichos resultados y a través de un Chain Classifier.

Para la media aritmética se recuperan las predicciones de los conjuntos de test de cada uno de los clasificadores, Decision Tree y Random Forest. La estructura de estas predicciones viene fija por las probabilidades individuales de que cada canción corresponda a uno de los seis posibles géneros.

	prob_0	prob_1	prob_2	prob_3	prob_4	prob_5	tag
0	0.128959	0.144796	0.366516	0.022624	0.124434	0.212670	4
1	0.002567	0.027464	0.137064	0.002053	0.001027	0.829825	5
2	0.734818	0.010562	0.123482	0.003785	0.053072	0.074283	0
3	0.018440	0.041135	0.048227	0.573050	0.300709	0.018440	3
4	0.610331	0.010030	0.181545	0.003009	0.054162	0.140923	0
...
60439	0.734818	0.010562	0.123482	0.003785	0.053072	0.074283	1
60440	0.043360	0.046070	0.111111	0.167480	0.580488	0.051491	4
60441	0.016966	0.037924	0.092814	0.551896	0.276447	0.023952	3
60442	0.640507	0.009253	0.151474	0.010624	0.040781	0.147361	5
60443	0.149623	0.079656	0.358450	0.012917	0.210980	0.188375	2

Figura 4.14. Predicciones individuales Decision Tree

Por ejemplo, para el caso de la primera canción, el Decision Tree devuelve la distribución de probabilidades marcada en azul, con 2 (Género Pop) como predicción final; errando frente a la clase real 4 (Género Rhythm and Blues).

La precisión obtenida en test por este clasificador era de 51.10%. Al sumarle y aplicarle la media aritmética con las predicciones del CNN 1D, 52.41% en test, se obtiene una precisión final de **55.39%**. Mejorando en torno a un 3% las métricas individuales.

Se repite el estudio con el resto de combinaciones de nuestros modelos.

Tabla 4.3. Resultados Ensemble por Media Aritmética

	ACC. TEST	TIEMPO
Decision Tree	51.10%	9min 36s
Random Forest	55.13%	2h 26min 6s

CNN 1D	52.41%	2h 17min 23s
LSTM	51.32%	8h 38min 15s

Decision Tree + CNN 1D	55.39%	6s
Decision Tree + LSTM	57.37%	6s
Random Forest + CNN 1D	59.03%	6s
Random Forest + LSTM	59.30%	6s

Resulta interesante ver cómo la combinación que a priori podría ser la mejor en términos de accuracy, Random Forest combinado con Red Neuronal Convolucional de una Dimensión, queda por debajo de la que combina Random Forest con el modelo de Long-short Term Memory.

A continuación, se prueba la técnica de Model Stacking. Se combinan las salidas de ambos modelos base en un dataset que será procesado por un tercer modelo, meta modelo.

El meta modelo utiliza enfoques algorítmicos para combinar las predicciones de los modelos base de forma ponderada. La ponderación se basa en la confianza o la habilidad relativa de cada modelo para hacer predicciones precisas en cada conjunto de datos. La idea clave es que, al combinar las predicciones, los errores individuales de los modelos se cancelen mutuamente o se reduzcan, lo que resultará en una predicción más precisa y robusta.

Para nuestro estudio se utilizará un modelo de Regresión Logística como meta modelo. Se fijará el máximo número de iteraciones del algoritmo para converger a **max_iter = 5000** y el resto de hiperparámetros serán optimizados a través de una búsqueda exhaustiva CV:

- **C = [10e-3, 10e-2, 10e-1, 1, 10, 100, 1000]**
- **class_weight = [None, 'balanced']**

El parámetro de regularización **C**, marcará la fuerza de regularización. Es decir, limitará en mayor o menor medida la influencia de los coeficientes de las variables predictoras. Por otro lado, **class_weight** permitirá al algoritmo ajustar automáticamente los pesos de las decisiones en torno a la frecuencia de la clase o no.

Tras su entrenamiento el meta modelo devuelve la precisión final por cada combinación de modelos base.

Tabla 4.4. Resultados Ensemble por Model Stacking

	ACC. TEST	TIEMPO
Decision Tree	51.10%	9min 36s
Random Forest	55.13%	2h 26min 6s

CNN 1D	52.41%	2h 17min 23s
LSTM	51.32%	8h 38min 15s

Decision Tree + CNN 1D [MS]	58.51%	25min 44s
Decision Tree + LSTM [MS]	58.08%	30min 7s
Random Forest + CNN 1D [MS]	60.37%	22min 1s
Random Forest + LSTM [MS]	60.04%	14min 27s

Se observa una mejora de alrededor de 1% de accuracy en prácticamente todas las combinaciones. Demostrando que la técnica de Model Stacking pondera de manera más efectiva las predicciones individuales que el caso básico de la Media Aritmética.

La combinación que mejor se comporta para este ensemble y para el estudio hasta el momento es la de Random Forest con Red Neuronal Convolutacional Unidimensional, con un **60.37%** de accuracy.

Por último, se probará el enfoque de los modelos Chain Classifier. Este tipo de arquitecturas está basado en el procesamiento de salidas de un modelo como entradas del siguiente. Es decir, las características o “conclusiones” obtenidas por uno de los modelos de nuestra arquitectura alimentarán el input de otro.

En este caso, se agregarán al conjunto de Audio Features las predicciones obtenidas para la parte de la lyrics. La intuición tras esto es que sirvan como características de audio adicionales por las que puedan entrenar el Decision Tree y el Random Forest.

Tabla 4.5. Resultados Ensemble por Chain Classifier

	ACC. TEST	TIEMPO
Decision Tree	51.10%	9min 36s
Random Forest	55.13%	2h 26min 6s

CNN 1D	52.41%	2h 17min 23s
LSTM	51.32%	8h 38min 15s

CNN 1D + Decision Tree [CC]	56.26%	9min 30s
LSTM + Decision Tree [CC]	55.79%	9min 16s
CNN 1D + Random Forest [CC]	61.10%	1h 55min 24s
LSTM + Random Forest [CC]	61.03%	1h 45min 53s

La mejor solución en términos de accuracy para el Chain Classifier es la que combina las predicciones de la Red Convolucional con la arquitectura Random Forest, superando casi en un 1% al Model Stacking de mismos modelos base. Sin embargo, hay que tener en cuenta el factor tiempo de ejecución, ya que para una mejora de alrededor del **0.7%** de accuracy, la técnica de Chain Classifier ha tardado un **81%** más.

Capítulo 5

Conclusiones y Trabajo Futuro

5.1 Conclusiones

En esta tesis, se abordó el desafío de clasificar géneros musicales utilizando tanto las letras de las canciones como las características de audio. El objetivo principal era desarrollar un modelo de ciencia de datos que pudiera aprender patrones y relaciones entre los datos para predecir de manera efectiva el género musical de una canción.

Para lograr este objetivo, se recopiló un conjunto de datos que incluía letras de canciones y características de audio de una amplia variedad de géneros musicales. Se utilizó un enfoque de preprocesamiento exhaustivo para limpiar y normalizar los textos de las letras, y se extrajeron características de audio relevantes como el tempo, la energía, el tono y la melodía.

Gracias al enriquecimiento de la base de datos original se pudo tener un contexto más preciso de las canciones. Algunas de las características extraídas como la energía daban un enfoque discriminante para géneros como rock, con ritmos generalmente enérgicos. Además, el índice de speechiness o hablabilidad también sirvió para dar contexto a la etiqueta miscellaneous. Muchas de las pistas almacenadas como miscellaneous resultaron ser obras narradas en formato audiolibro.

La extracción de características de audio también ayudó a la creación de modelos de aprendizaje automático más complejos. Resultando en una división de la investigación en dos partes: soluciones de Machine Learning para las características de audio y soluciones de Deep Learning para las letras de las canciones.

Se obtuvieron resultados competentes en ambas partes, concluyendo que el mejor algoritmo para la parte de Machine Learning resulta ser el Bosque Aleatorio y para la de Deep Learning, la Red Convolucional Unidimensional. Por otro lado, en términos de tiempo de ejecución, la Red Convolucional reporta también los mejores resultados, mientras que el Bosque Aleatorio se ve superado por el Árbol de Decisión.

Finalmente se combinaron los modelos de ambas partes con el objetivo de obtener un meta modelo final más completo.

La métrica de tiempo de ejecución resulta muy útil para obtener conclusiones finales. Quizá el modelo con mejor tasa de acierto es el Chain Classifier compuesto por Red Convolucional Unidimensional y Bosque Aleatorio (61.10 %), sin embargo, al comparar con la media aritmética de los modelos Árbol de Decisión y Red Convolucional (55.39 %) se observa que para aumentar un 5% la precisión del modelo final se ha necesitado aumentar en algo más 4 horas el entrenamiento.

En términos generales, se concluye que la clasificación de géneros musicales a partir de las letras de las canciones y las características de audio es un problema complejo pero abordable mediante técnicas de aprendizaje automático. El modelo desarrollado en esta tesis proporciona una base sólida para futuras investigaciones y aplicaciones en el campo de la recomendación musical, la creación de listas de reproducción y la segmentación de audiencias musicales.

Sin embargo, es importante destacar que existen algunas limitaciones en este enfoque. La calidad y disponibilidad de los datos pueden variar, lo que puede afectar la precisión de las predicciones. Además, la clasificación de géneros musicales es un concepto subjetivo y puede haber discrepancias entre diferentes oyentes y expertos en la industria musical.

5.2 Trabajo futuro

Para que el presente trabajo perduró y pueda servir como base para futuras investigaciones resulta de vital importancia ir actualizando los datos y con ello los modelos clasificadores.

Es importante actualizar constantemente los modelos de clasificación de géneros musicales debido a varios factores. En primer lugar, la música es un arte en constante evolución y los nuevos estilos y subgéneros surgen regularmente. La aparición de nuevos géneros musicales puede ser el resultado de innovaciones tecnológicas, cambios en las tendencias culturales o la influencia de diferentes tradiciones musicales. Por lo tanto, actualizar los modelos de clasificación permite mantenerse al día con las expresiones musicales contemporáneas y reconocer las nuevas formas de creatividad.

En segundo lugar, los géneros musicales a menudo se entrelazan y se mezclan entre sí. La música contemporánea es cada vez más interdisciplinaria y fusiona elementos de diferentes estilos y tradiciones musicales. Esto hace que la clasificación estricta en categorías rígidas sea cada vez más difícil y limitada. Actualizar los modelos de clasificación permite capturar estas fusiones y reconocer la diversidad y la complejidad de la música actual.

Además, los modelos de clasificación de géneros musicales pueden estar influenciados por sesgos y estereotipos culturales. Al actualizar los modelos, se puede trabajar en eliminar estos sesgos y promover una clasificación más inclusiva y equitativa. La música es una forma de expresión cultural y artística que trasciende las barreras sociales y culturales, y la clasificación de géneros musicales debe reflejar esta diversidad y promover la inclusión.

Por último, la actualización de los modelos de clasificación de géneros musicales tiene un impacto en diversas áreas, como la industria musical, la investigación académica y la recomendación de música en plataformas de streaming. Un modelo de clasificación actualizado y preciso puede ayudar a los artistas a encontrar su audiencia y promover su música, facilitar la investigación sobre la evolución de los estilos musicales y ofrecer recomendaciones de música más personalizadas y relevantes para los oyentes.

Por otro lado, al analizar las características de audio que se extrajeron de Spotify se observó que una de ellas era “speechiness” (índice hablabilidad o presencia de elementos hablados en una canción). Quizá resultaría interesante darle más importancia dentro de nuestro estudio por estar directamente relacionada con las letras de las canciones. A mayor índice de speechiness mayor probabilidad de encontrar contenido hablado dentro de una pista y por tanto mayor cantidad de palabras dentro de una lyrics.

El término "speechiness" se refiere a una medida que se utiliza para evaluar el grado de predominio del habla en una grabación de audio, especialmente en el contexto de la música. Esta métrica cuantifica la proporción de contenido hablado en comparación con la cantidad total de sonidos en una pista de audio. La hablabilidad se calcula analizando diversas características acústicas del audio, como la energía en frecuencias asociadas con el habla, la presencia de pausas y silencios, y la forma en que se distribuyen las amplitudes a lo largo del tiempo. Estas características se utilizan para inferir la presencia de contenido hablado en la grabación.

En relación con la cantidad de palabras en una letra (lyrics), la speechiness puede estar correlacionada. Por lo general, si una canción tiene un mayor número de palabras en su letra, es más probable que contenga una proporción significativa de contenido hablado en comparación con una canción con menos palabras.

Por este motivo parece interesante valorar un modelo que dependiendo del índice de speechiness de mayor o menor prioridad a la predicción del modelo de procesamiento natural de las letras.

Pero, sin duda, la principal tarea de trabajo futuro ha sido el despliegue del modelo final. Hasta ahora en este proyecto se ha analizado el contexto de la clasificación de géneros musicales, se han desarrollado y evaluado modelos para abordar este problema y se han combinado los mejores modelos para obtener un meta modelo capaz de resolver el problema en gran medida. Sin embargo, se ha quedado pendiente la implementación práctica del modelo dentro de un sistema o aplicación real. Pasar del entorno de desarrollo a un entorno de producción real donde dar a conocer sus funcionalidades a los usuarios.

El despliegue de este tipo de modelo facilita la búsqueda y recomendación de música para los usuarios. Al asignar géneros a las canciones, se establece una estructura que ayuda a los sistemas de recomendación a sugerir nuevos contenidos a los oyentes, basándose en sus preferencias y gustos previos. Esto mejora la experiencia del usuario y aumenta la posibilidad de descubrir nuevas canciones y artistas que se ajusten a sus intereses.

Además, el despliegue de un modelo de clasificación de géneros musicales puede ser beneficioso para los creadores y la industria de la música en general. Los artistas pueden utilizar esta clasificación para obtener información sobre las tendencias musicales y la popularidad de diferentes géneros en un determinado momento. Esto les permite adaptar su música a las preferencias del público y tomar decisiones informadas sobre sus futuras producciones.

Para las discográficas y plataformas de streaming, contar con un modelo de clasificación de géneros les ayuda a organizar y categorizar su vasto catálogo de música. Esto facilita la gestión y el etiquetado de las canciones, lo que a su vez optimiza los algoritmos de recomendación y permite ofrecer a los usuarios una experiencia musical más personalizada y satisfactoria.

En resumen, desplegar un modelo de clasificación de géneros musicales es esencial en la actualidad para mejorar la organización, búsqueda y recomendación de música. Además, beneficia tanto a los oyentes, quienes descubren nueva música acorde a sus preferencias, como a los artistas y la industria de la música en general, al proporcionar información valiosa sobre las tendencias y la popularidad de diferentes géneros.

Bibliografía

- Tsapsinos, D., Kehagias, D., Koutentakis, G., & Kotropoulos, C. (2016). Music genre classification based on audio and lyrics. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 551-555).
- Li, Y., Yang, Y., & Zhou, Y. (2018). Music genre classification based on audio and lyrics using deep neural network ensemble. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management (pp. 1361-1370).
- Han, J., Dong, X., & Liu, X. (2020). A multi-modal approach to music genre classification with lyrics and audio. In 2020 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1-6).
- Cao, Y., Zhang, X., Lu, Z., & Xie, C. (2021). A hybrid approach for music genre classification combining acoustic features and lyrics. *Journal of Intelligent Information Systems*, 56(1), 75-93.
- Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees*. CRC Press.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 1(1), 81-106.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, 2(3), 18-22.

Cutler, D. R., Edwards Jr, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology*, 88(11), 2783-2792.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Zhang, Y., & LeCun, Y. (2015). Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).

Graves, A. (2012). Supervised sequence labelling with recurrent neural networks (Doctoral dissertation). Technical University of Munich.

- Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 6645-6649). IEEE.
- Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10), 2451-2471.
- Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*

Anexo I. Desarrollo Realizado

I.1 Enriquecimiento de la Base de Datos

- [spotify_script.ipynb](#)
- [read spotify scrapping results.ipynb](#)

I.2 Machine Learning

- [5M Songs – Audio features EDA + DT](#)
- [5M Songs – Audio features EDA + RF](#)

I.3 Deep Learning

- [5M Songs – GloVe + CNN](#)
- [5M Songs – GloVe + LSTM](#)

I.4 Ensemble

1. Model Stacking

- [5MS – Model Stacking \(Tree + CNN\)](#)
- [5MS – Model Stacking \(Tree + LSTM\)](#)
- [5MS – Model Stacking \(Forest + CNN\)](#)
- [5MS – Model Stacking \(Forest + LSTM\)](#)

2. Chain Classifier

- [5MS – Chain Classifier \(CNN + Tree\)](#)
- [5MS – Chain Classifier \(LSTM + Tree\)](#)

- [5MS – Chain Classifier \(CNN + Forest\)](#)
- [5MS – Chain Classifier \(LSTM + Forest\)](#)