INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Unidad de aprendizaje
Reconocimiento de voz

Proyecto
Análisis de sentimiento

Alumno
Jose Octavio Rios Pérez

Grupo
7BM1

1. Introducción

Propósito del Código: El código está diseñado para preprocesar y analizar archivos de audio. Extrae características como MFCCs, tasa de cruce por cero, y tono (pitch), y luego realiza análisis estadísticos y de reducción de dimensionalidad sobre estas características para después estas características entrenarlos en diferentes modelos de machine learning

2. Detalles del Código

Lenguaje de Programación: Python 3.11.5.

Librerías Utilizadas: librosa, numpy, matplotlib, pandas, sklearn.

Dataset utilizado: RAVDESS Emotional speech audio(https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio)

CSV generado: datos_voz

Funcionalidades Principales:

- Extracción de características de audio como MFCCs, tasa de cruce por cero y pitch.
- Análisis estadístico de estas características.
- Reducción de la dimensionalidad con PCA.
- Creación de un archivo CSV con las características extraídas.

Extracción de Características:

- **MFCCs (Coeficientes Cepstrales de Frecuencias Mel):** Se extraen 40 MFCCs, que son ampliamente utilizados en el procesamiento de señales de audio, especialmente para el análisis de voz.
- **Tasa de Cruce por Cero (ZCR):** Mide la tasa a la que la señal cambia de positivo a negativo o viceversa. Este es un indicador importante para identificar la naturaleza rítmica de un sonido.
- **Pitch (Tono):** Se extrae utilizando el algoritmo YIN, una técnica común para la estimación de la frecuencia fundamental en señales de voz.

Procesamiento de Datos:

- **Normalización:** Uso de StandardScaler para normalizar las características, lo que es crucial para modelos de aprendizaje automático y análisis estadístico.
- **Reducción de Dimensionalidad:** Uso de PCA para reducir la dimensionalidad de las características MFCCs, ayudando a identificar las componentes más importantes.

Entrenamiento con ML:

- SVM (Máquinas de Vectores de Soporte): Se emplea con kernel RBF. Se evalúa la precisión usando validación cruzada.

- Random Forest: Un modelo basado en ensamble de árboles de decisión. Se evalúa de manera similar a SVM.
- XGBoost (Gradient Boosting Machine): Un modelo avanzado de boosting que es popular por su rendimiento y velocidad.

Modelado y Evaluación:

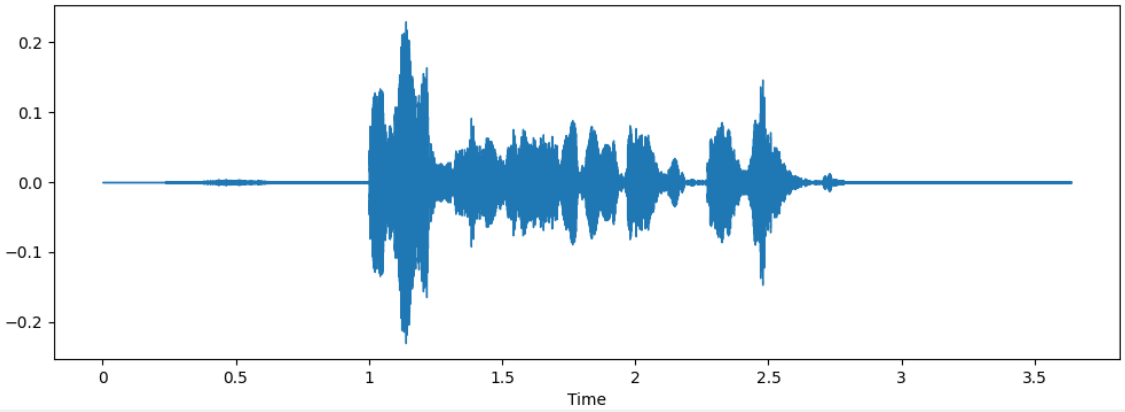
- **Carga de Datos:** Los datos se cargan desde un archivo CSV, lo que implica que el preprocesamiento previo almacenó las características de audio en este formato.
- **Preparación de Datos:** Se convierten las etiquetas categóricas a numéricas usando LabelEncoder, preparándolas para el modelado.
- **División de Datos:** Aunque no se ejecuta en el código proporcionado, se prepara una división de datos de entrenamiento y prueba (comentada).
- **Validación Cruzada:** Uso de StratifiedKFold para mantener un balance de clases, lo que es crucial en datasets desequilibrados.

3. Análisis del Código

Robustez en la Extracción de Características: El código aborda varios aspectos clave del análisis de voz, como los MFCCs y el tono, lo que lo hace robusto para aplicaciones de análisis de audio.

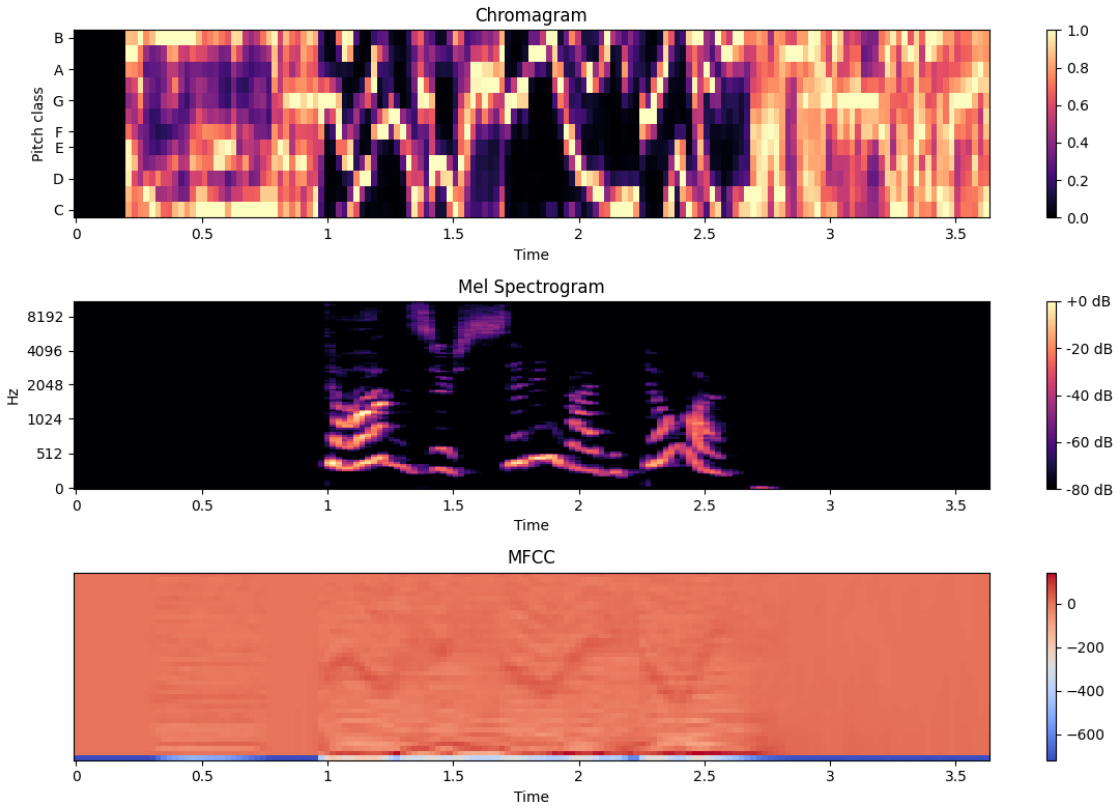
Preprocesamiento Avanzado: La inclusión de pasos como la normalización y la reducción de la dimensionalidad indica un enfoque sofisticado hacia el procesamiento de datos de audio.

Ahora analizaremos diferentes graficas para entender los datos y la representación de las características.



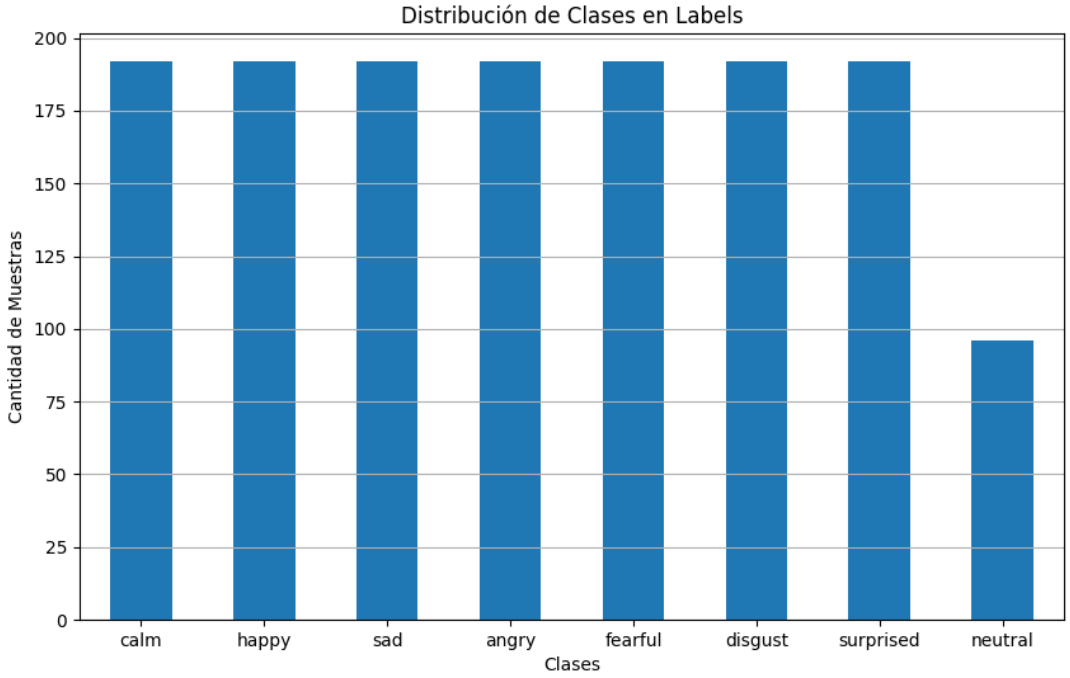
visualización general de los audios A

Análisis de cronograma, espectograma y MFCC’s



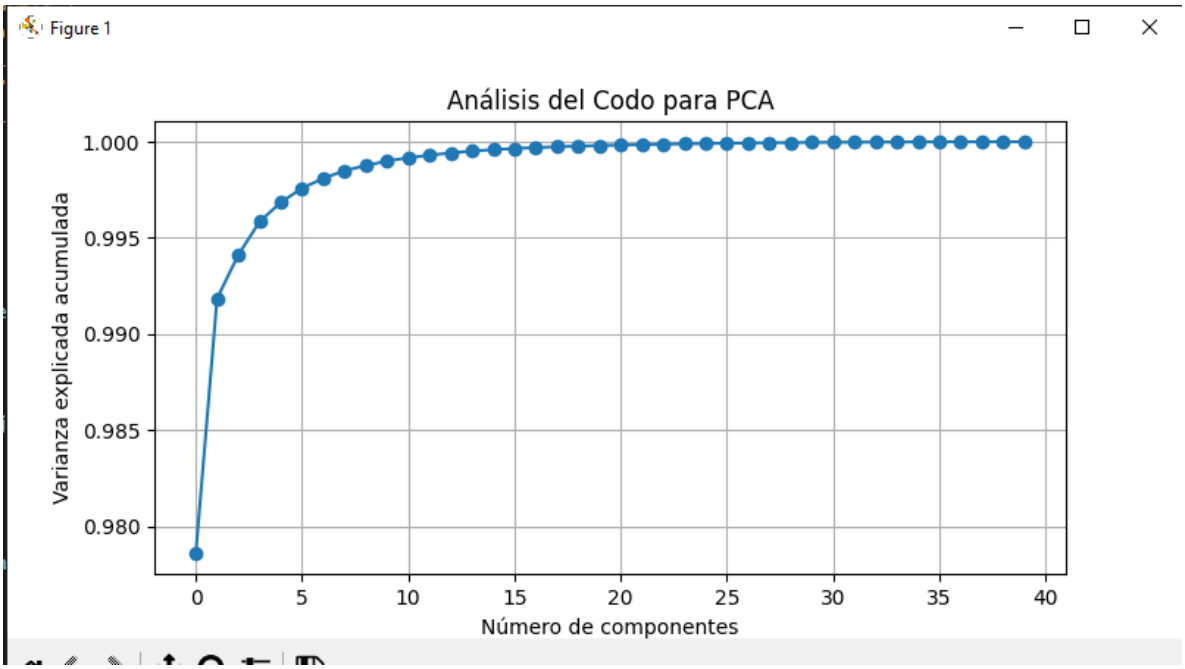
visualización de características B

Distribución de las etiquetas del dataset



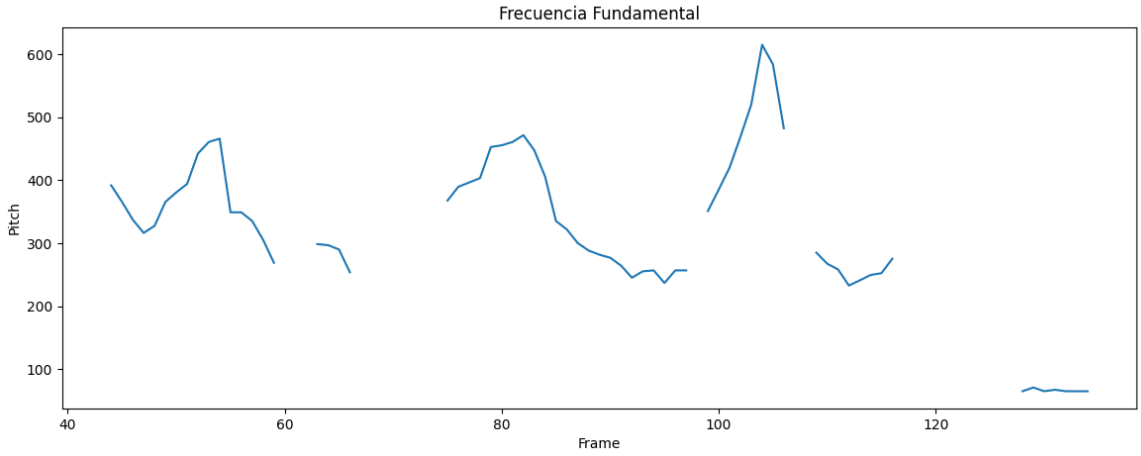
visualización de la distribución de etiquetas C

Análisis del codo para determinar el numero optimo de componentes para el algoritmo PCA



visualización de codo de jambú D

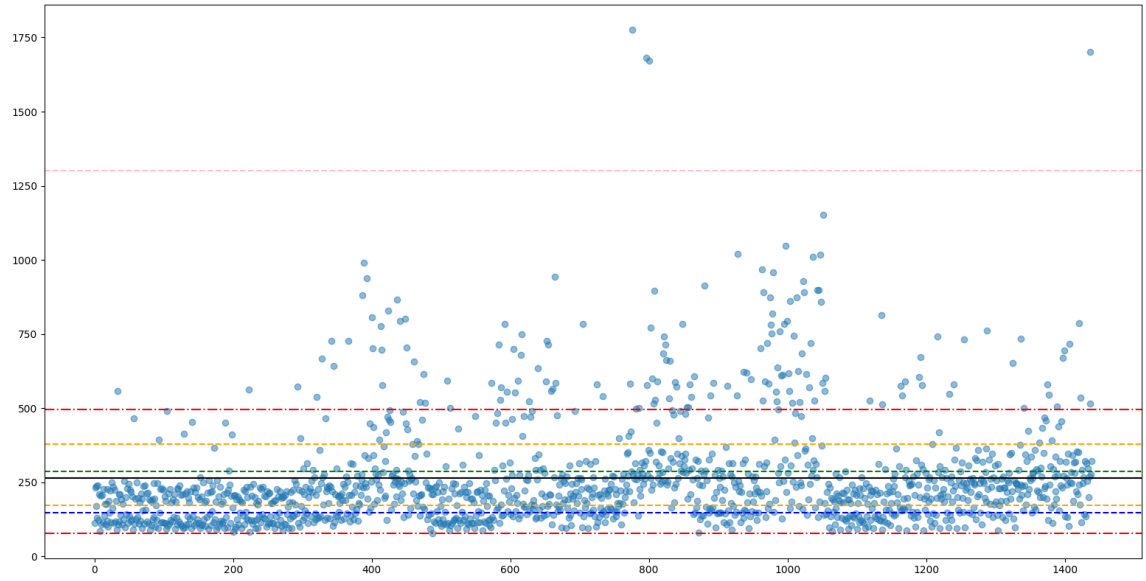
Grafica que muestra el donde se encontró un pitch en un audio



visualización del pitch E

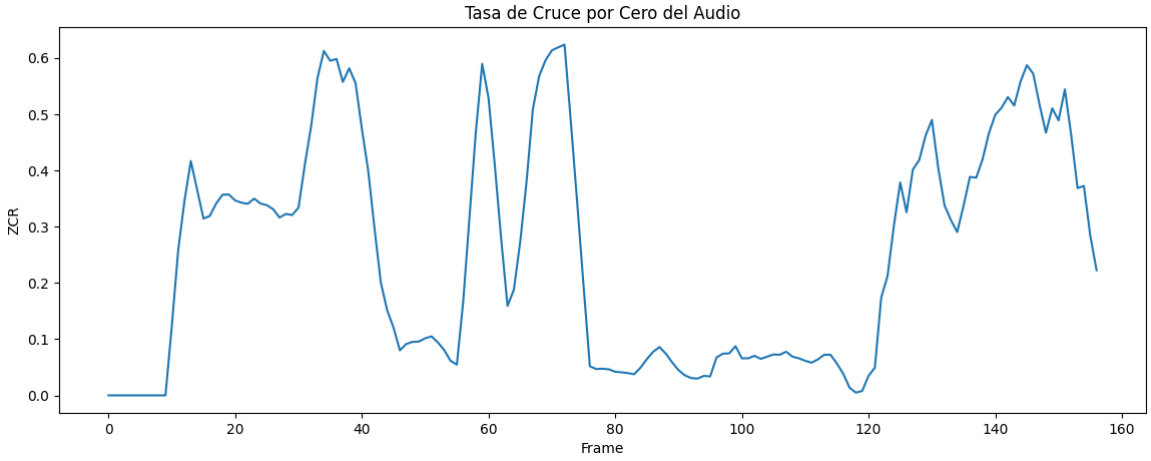
grafico que nos da la mediana del pitch de los datos.

Las líneas rojas y negra son limites que nos ayudaran a crear una tabla de referencia



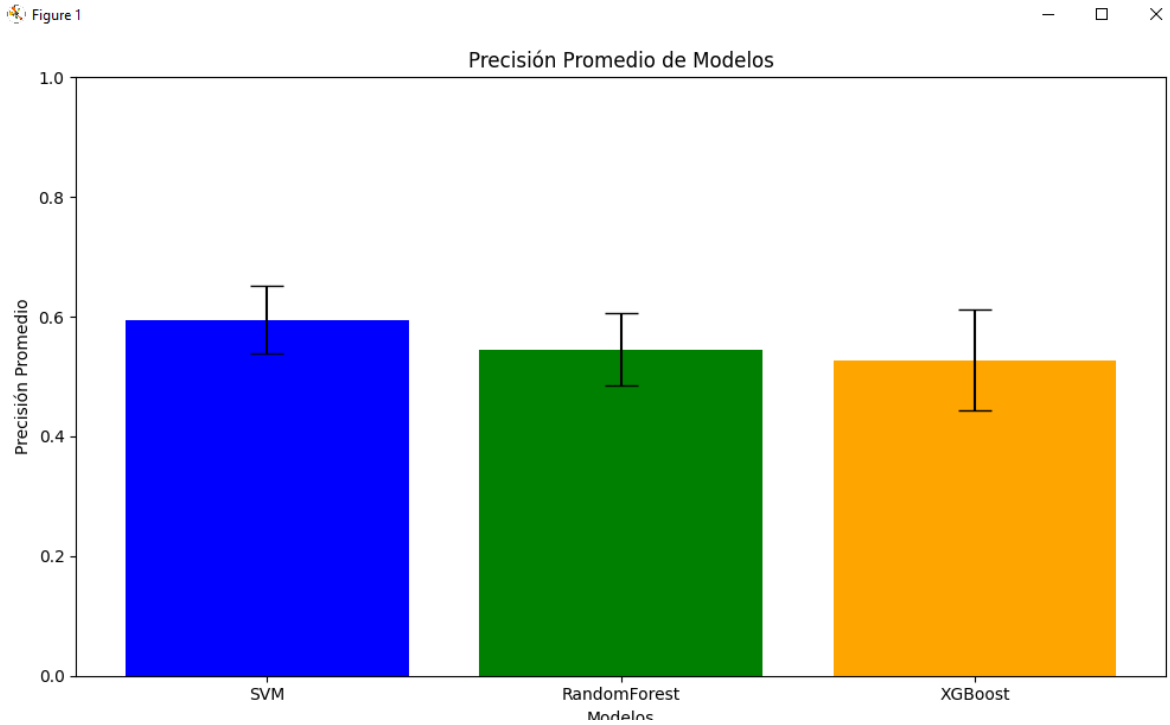
visualización de la mediana del pitch F

Visualización de cruce por cero en un audio o el cambio de negativo a positivo o viceversa del tono del audio.



visualización Cruce por Cero G

Grafica que representa la precisión promedio y la confianza de los modelos



visualización de los modelos H

- **Selección de Modelos:** El código utiliza tres modelos diferentes, lo que permite comparar su eficacia en el dataset.
- **Evaluación de Modelos:** Se realiza una evaluación sistemática mediante la validación cruzada, proporcionando una estimación robusta del rendimiento del modelo.

4. Casos de Uso y Ejemplos

Análisis de Emociones: Con la extracción de etiquetas de sentimiento e intensidad de los nombres de archivo, el código puede ser útil en el campo del análisis de emociones a partir de la voz.

Clasificación de Emociones en Voz: Con la inclusión de modelos de aprendizaje automático, el código puede ser utilizado para clasificar emociones o características en grabaciones de voz.

5. Problemas y Limitaciones

Generalización: El código está diseñado específicamente para un tipo de archivo de audio y estructura de datos. Puede requerir modificaciones para otros formatos o estructuras.

Manejo de Errores: Hay poca gestión de errores para situaciones como archivos de audio corruptos o formatos incompatibles.

Dependencia de la Estructura del Nombre de Archivo: La extracción de etiquetas de sentimiento y de intensidad depende de la nomenclatura específica de los archivos de audio, lo que podría limitar su aplicación a otros conjuntos de datos sin esta estructura.

Rendimiento del Modelo: Basándose en los resultados, los modelos muestran una precisión moderada. Esto podría indicar la necesidad de un preprocesamiento más avanzado, una selección de características más detallada, o un ajuste de hiperparámetros.

6. Mejoras Sugeridas

Optimización: Implementar técnicas de paralelización para el procesamiento de grandes conjuntos de datos.

Flexibilidad en la Extracción de Etiquetas: Modificar el código para permitir una mayor flexibilidad en la forma en que se extraen las etiquetas de los archivos, permitiendo su uso en diferentes conjuntos de datos.

Exploración de Otros Modelos: Probar con otros modelos de aprendizaje automático o técnicas de deep learning.

7. Conclusión

Resumen: El código es una herramienta avanzada para el análisis de audio, útil en el análisis de emociones y en la caracterización de señales de voz. Las técnicas de preprocesamiento y extracción de características son robustas y adecuadas para una variedad de aplicaciones.

Además representa un flujo de trabajo integral para el análisis de datos de audio, desde la extracción de características hasta el modelado predictivo. Sin embargo, hay margen para mejorar la precisión de los modelos.

Resultados:

Resultados de SVM:

> *Puntajes de precisión de cada partición: [0.60416667 0.55555556 0.58333333 0.61805556 0.58333333 0.58333333*
>
> *0.625 0.56944444 0.65277778 0.57638889]*

Precisión promedio: 0.60 (+/- 0.06)

Resultados de RandomForest:

Puntajes de precisión de cada partición: [0.50694444 0.54166667 0.59722222 0.54861111 0.5 0.52083333

0.5625 0.53472222 0.59027778 0.54861111]

Precisión promedio: 0.55 (+/- 0.06)

Resultados de GBM (XGBoost):

Puntajes de precisión de cada partición: [0.52777778 0.43055556 0.59027778 0.54166667 0.52083333 0.51388889

0.56944444 0.51388889 0.5625 0.50694444]

Precisión promedio: 0.53 (+/- 0.08)

8. Recomendaciones Finales

- **Implementación de Mejoras:** Se recomienda implementar las mejoras sugeridas para aumentar la utilidad y robustez del código.
- **Pruebas Adicionales:** Realizar pruebas con diferentes conjuntos de datos para evaluar la eficacia y precisión del análisis.
- **Mejora Continua:** Continuar la experimentación con diferentes enfoques y técnicas para mejorar la precisión de los modelos de clasificación.

Explicación de la funciones:

preprocessing(audio_path):

- Propósito: Procesar un archivo de audio y extraer varias características de él.
- Pasos Principales:
    - Carga un archivo de audio usando librosa.load.
    - Calcula el tamaño de la ventana para el análisis.
    - Extrae características de MFCC (Coeficientes Cepstrales de Frecuencia Mel) del audio.
    - Calcula la media de los MFCCs.
    - Extrae y calcula la media de la Tasa de Cruce por Cero (Zero Crossing Rate, ZCR).
    - Extrae el tono (pitch) del audio y calcula su mediana.
    - Divide el nombre del archivo para obtener información sobre el sentimiento e intensidad.
    - Agrega las características extraídas a listas para análisis posteriores.

bamboo_elbow_method(pca, mfccs):

- Propósito: Utilizar el método del codo para determinar el número óptimo de componentes principales en un análisis PCA (Principal Component Analysis) aplicado a MFCCs.
- Pasos Principales:
    - Aplica PCA a los MFCCs.
    - Calcula y grafica la varianza explicada acumulada para identificar el "punto de codo".
    - analize_pitch(pitch_median_array, median_global_pitch):
    - Propósito: Analizar un conjunto de valores de tono (pitch) para identificar umbrales y valores atípicos utilizando el Rango Intercuartílico (IQR).
    - Pasos Principales:
    - Calcula los cuartiles y el IQR del array de tonos.
    - Establece umbrales para identificar valores atípicos.
    - Ajusta los umbrales para asegurarse de que estén dentro de rangos razonables.

create_csv(labels_array,zcr_array,intensity_array,pitch_median_array,frecuency_array,features_array):

- Propósito: Crear un archivo CSV a partir de varios arrays que contienen etiquetas y características extraídas del audio.
- Pasos Principales:
    - Verifica que todos los arrays tengan la misma longitud.
    - Crea DataFrames de pandas para cada conjunto de datos.
    - Concatena todos los DataFrames en uno solo.
    - Guarda el DataFrame final como un archivo CSV.