

T.P Identificador

1. Explicar el funcionamiento del código.
 2. Sacar a relucir los rasgos asociados a los distintos paradigmas de programación en el código.
 3. Explicar la función de cada script y cómo interactúan entre sí.
 4. Indicar los criterios y decisiones tomadas para el planteamiento del código.
 5. Explicar las fortalezas y debilidades del software diseñado
-

1. Explicación del Funcionamiento del Código

Este código utiliza Python para analizar un archivo de audio, estimar su frecuencia base y determinar si la voz pertenece a un hombre o a una mujer. El flujo general es el siguiente:

Carga de Configuración: Se leen los parámetros estadísticos (media y desviación estándar) para las frecuencias de voz masculina y femenina desde un archivo JSON.

Grabación de Audio: Se graba un archivo de audio de prueba utilizando la función `Grabar`.

Análisis del Audio:

Se calcula la frecuencia base del audio grabado con la clase `Audio`.

Se estiman las probabilidades condicionales de que la frecuencia base pertenezca a una voz masculina o femenina usando distribuciones normales.

Clasificación: Se determina el género en base a la probabilidad máxima y la fiabilidad del resultado.

Salida de Resultados: Se imprime la clasificación y el nivel de confianza en el resultado.

2. Rasgos Asociados a Paradigmas de Programación

Programación Procedimental:

El flujo está controlado mediante llamadas secuenciales a funciones como **Grabar** y **Audio**.

Se utilizan estructuras de control simples como bucles **for** y condicionales **if**.

Programación Orientada a Objetos:

La clase **Audio** encapsula funcionalidad para el análisis del audio (como el cálculo de la frecuencia base).

Programación Funcional:

Se emplean funciones puras, como **norm.pdf** para calcular la densidad de probabilidad de las distribuciones normales.

3. Explicación de los Scripts y su Interacción

Scripts Importados:

funciones_comunes:

Audio: Clase para manejar archivos de audio y calcular características como la frecuencia base.

Grabar: Función para capturar audio y guardarlo en un archivo.

Interacción:

El script principal inicia cargando configuraciones desde un archivo JSON.

Utiliza **Grabar** para capturar un archivo de audio temporal.

Usa **Audio** para analizar la frecuencia base del audio grabado.

Integra los datos estadísticos y el resultado del análisis del audio para calcular probabilidades con **norm.pdf** y determinar el género.

4. Criterios y Decisiones del Planteamiento del Código

Uso de JSON: Facilita la modificación y extensión de las configuraciones estadísticas sin alterar el código.

Distribuciones Normales: Modelan la probabilidad de que una frecuencia base pertenezca a una población masculina o femenina de manera intuitiva.

Criterios de Fiabilidad: Se garantiza que el resultado es confiable solo si las probabilidades son significativas (umbral de 0.0005) y suficientemente diferenciadas.

Iteración con Feedback: Se brinda retroalimentación al usuario en caso de resultados poco confiables, solicitando repetir la grabación.

5. Fortalezas y Debilidades del Software Diseñado

Fortalezas:

Modularidad: La separación en funciones y clases permite reusar y mantener el código fácilmente.

Interactividad: Incluye retroalimentación al usuario en caso de errores o datos insuficientes.

Extensibilidad: Configuraciones externas en JSON facilitan la adaptación a otros idiomas o pópulos con diferentes parámetros vocales.

Estadística Robustas: Uso de distribuciones normales para una estimación precisa.

Debilidades:

Dependencia de Ruido: El análisis puede ser afectado por ruido o grabaciones de baja calidad.

Simplificación de Género: Asume solo dos géneros basados en frecuencias vocales, lo que no cubre casos intermedios o excepciones.

Escalabilidad: Para bases de datos más grandes o configuraciones avanzadas, el enfoque podría volverse ineficiente.