Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Speech 2 Summarization - Actividad de evaluación NLP

TC3007C.501 Inteligencia Artificial Avanzada para la Ciencia de Datos II

Profesores:

Iván Mauricio Amaya Contreras

Blanca Rosa Ruiz Hernández

Félix Ricardo Botello Urrutia

Edgar Covantes Osuna

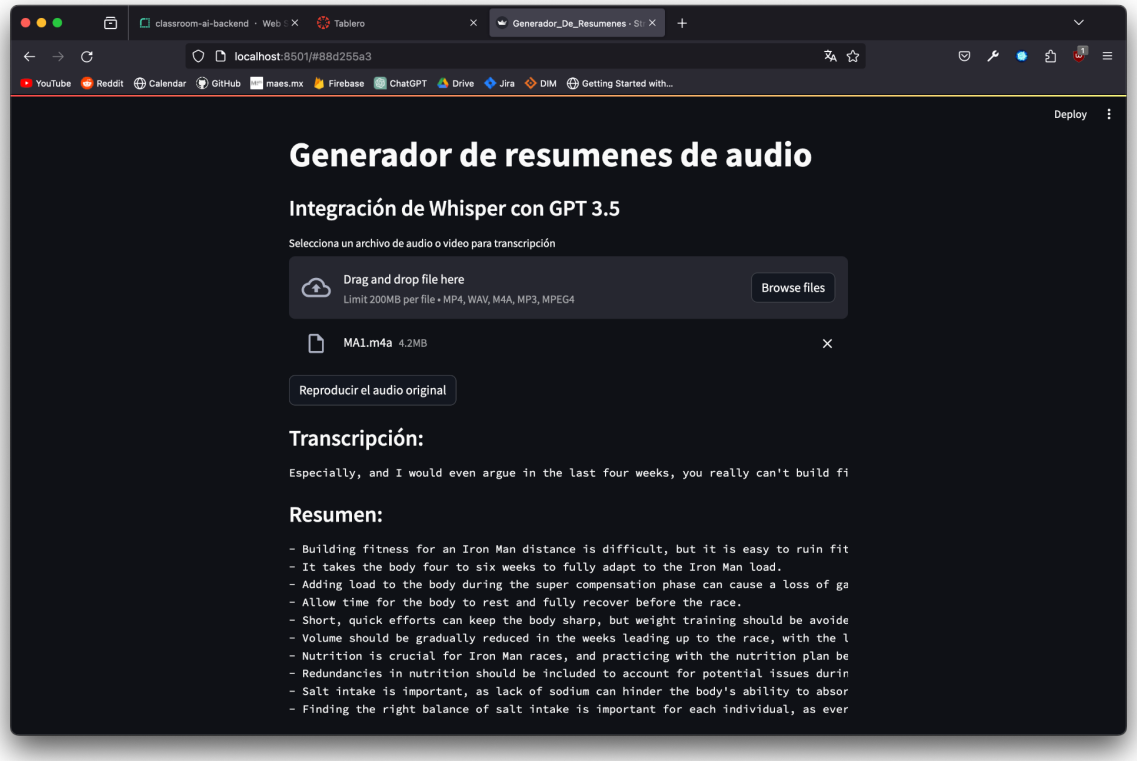
Felipe Castillo Rendón

Hugo Terashima Marín

Alumno:

Luis Ángel Guzmán Iribe - A01741757

24 de Noviembre de 2023

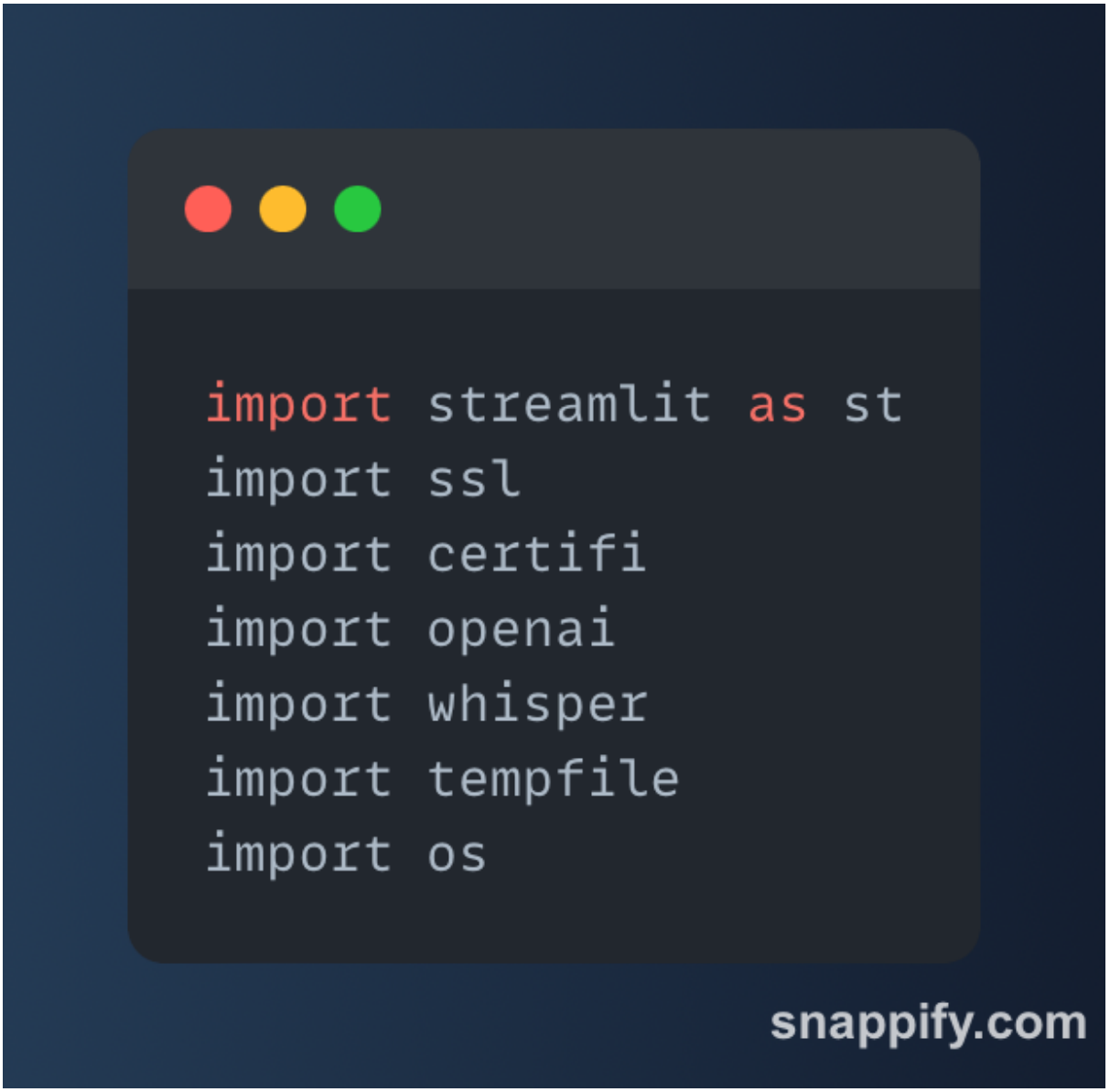


Este documento corresponde a la evidencia del módulo de Natural Language Processing, este programa sencillo consiste en una interfaz web implementada con Streamlit que permite interactuar con la API de OpenAI para ejecutar un servicio de transcripción de audio a texto, y posteriormente generar un resumen de dicha transcripción.

Para esto se emplean 2 servicios, empezando por Whisper, que toma como entrada un archivo de audio, analiza el lenguaje del mismo, extrae el contenido del archivo como texto, incluyendo señales de puntuación y demás pausas ortográficas.

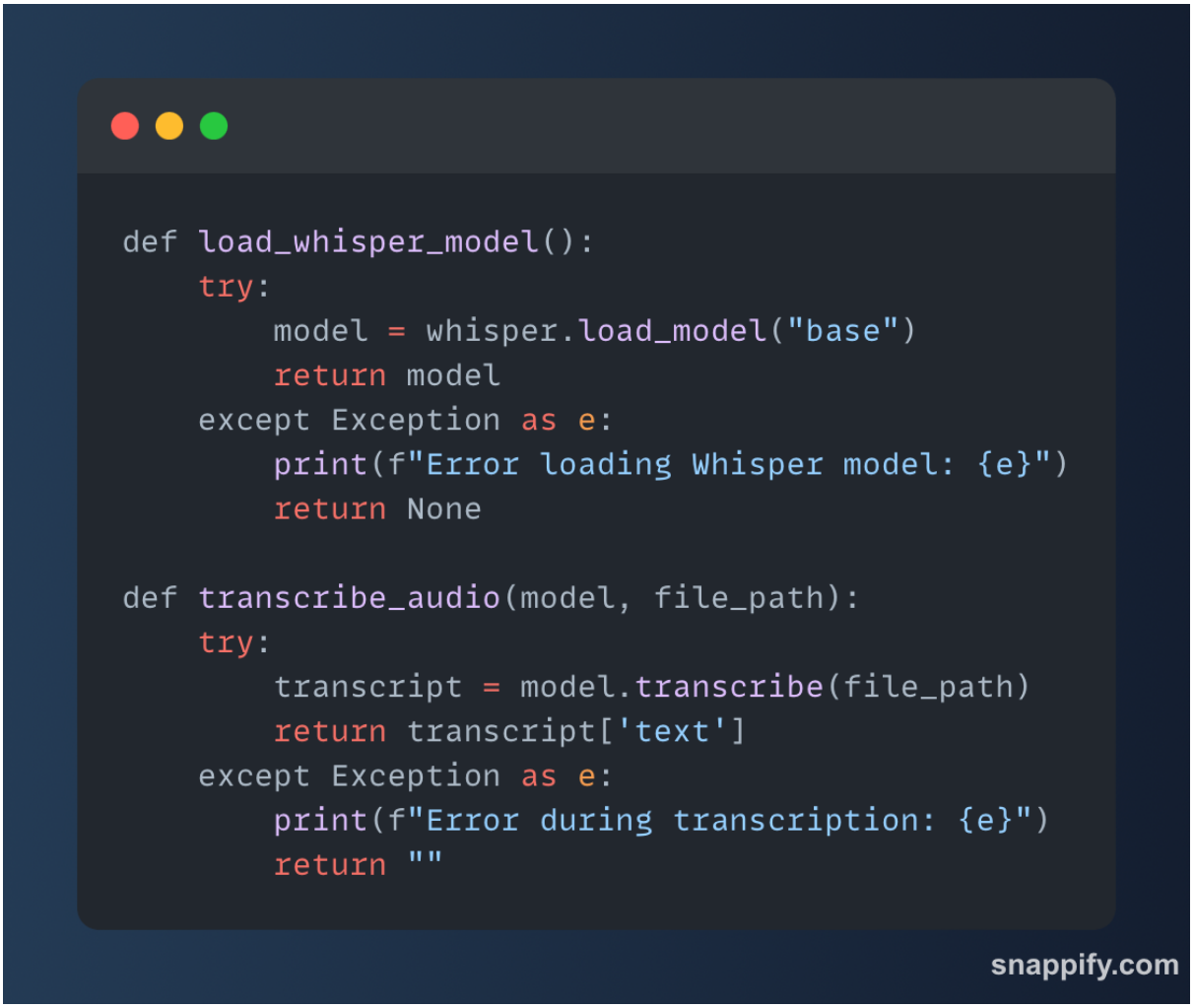
Una vez que se ha extraído el texto se utiliza el modelo 'gpt-3.5-turbo' (una versión diferente del modelo empleado por Chat GPT) se realiza una consulta sobre el texto extraído bajo la siguiente prompt: 'You are an office administrator, summarize the text in key points'. Esto otorga como resultado precisamente lo que solicita la prompt, un listado de bulletpoints con los puntos más importantes del texto.

A continuación se muestran algunos snippets de código relevantes para entender el funcionamiento del código.



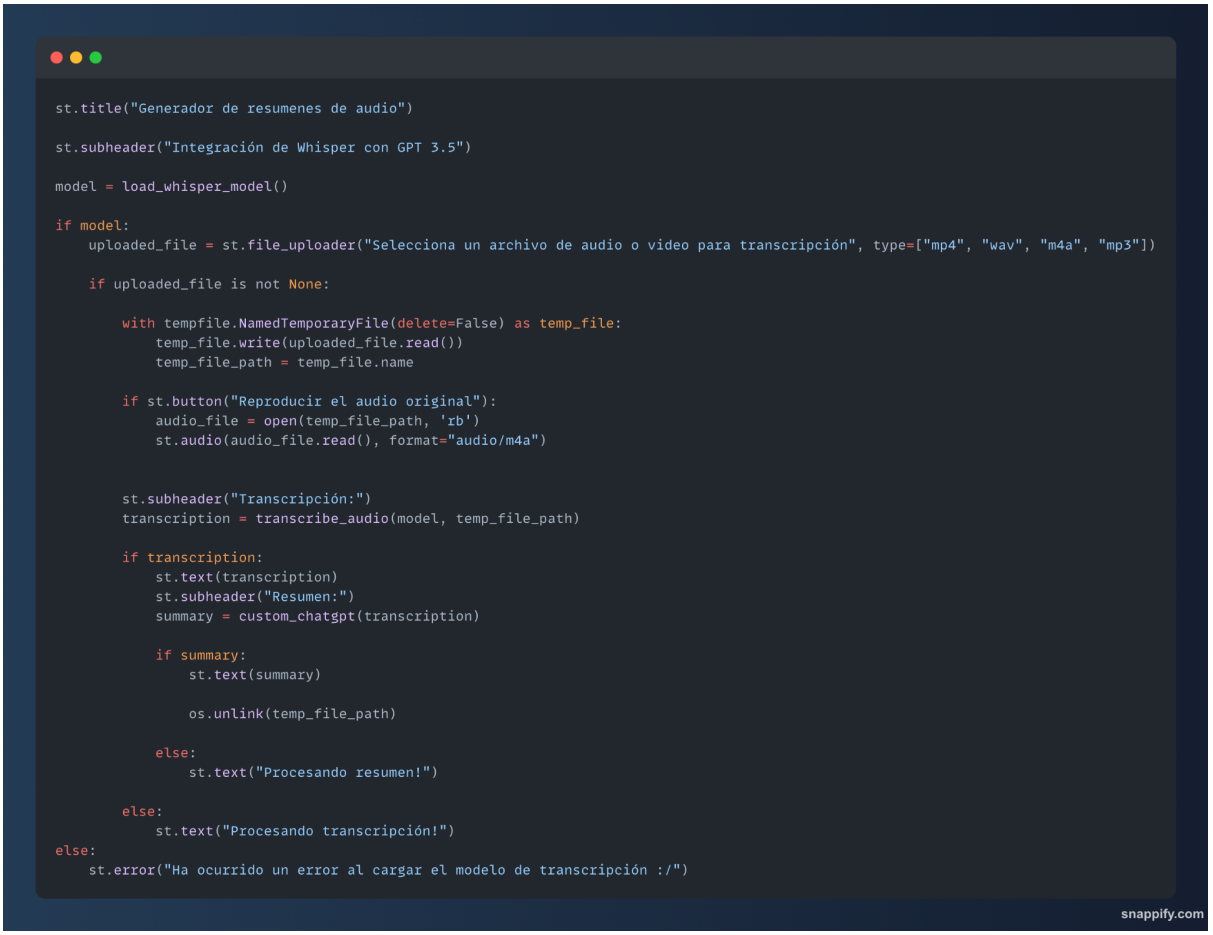
En esta imagen podemos ver las librerías empleadas para la creación de la interfaz. Sus usos son los siguientes:

- Streamlit: Se utiliza para generar una interfaz web sencilla que permite interactuar con los modelos de IA.
- SSL y Certifi: Se utilizan para generar un certificado SSL.
- OpenAI: API de Python de la compañía homónima que permite acceso al modelo GPT-3.5 Turbo
- Whisper: Permite acceso a la función de transcripción
- Temfile: Lectura del archivo de audio
- OS: Modificación de archivos en el sistema.



Se carga el modelo de Whisper con un wrapper para considerar potenciales errores.
Se emplea un Wrapper para correr la función de transcripción usando el modelo base de Whisper.

Función para llamar a GPT y realizar la consulta sobre el texto con la siguiente prompt: 'You are an office administrator, summarize the text in key points'



```python
st.title("Generador de resumenes de audio")

st.subheader("Integración de Whisper con GPT 3.5")

model = load_whisper_model()

if model:
    uploaded_file = st.file_uploader("Selecciona un archivo de audio o video para transcripción", type=["mp4", "wav", "m4a", "mp3"])

    if uploaded_file is not None:

        with tempfile.NamedTemporaryFile(delete=False) as temp_file:
            temp_file.write(uploaded_file.read())
            temp_file_path = temp_file.name

        if st.button("Reproducir el audio original"):
            audio_file = open(temp_file_path, 'rb')
            st.audio(audio_file.read(), format="audio/m4a")


        st.subheader("Transcripción:")
        transcription = transcribe_audio(model, temp_file_path)

        if transcription:
            st.text(transcription)
            st.subheader("Resumen:")
            summary = custom_chatgpt(transcription)

            if summary:
                st.text(summary)

                os.unlink(temp_file_path)

            else:
                st.text("Procesando resumen!")

        else:
            st.text("Procesando transcripción!")
else:
    st.error("Ha ocurrido un error al cargar el modelo de transcripción :/")
```


Código para que la interfaz de Streamlit interactúe con las funciones descritas ateriormente.