Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus, Monterrey

Inteligencia Artificial Avanzada para la Ciencia de datos - (Gpo 503)

Reporte

Generador de resúmenes utilizando Streamlit

Estudiante:

Samantha Guanipa Ugas A01703936

Equipo 4

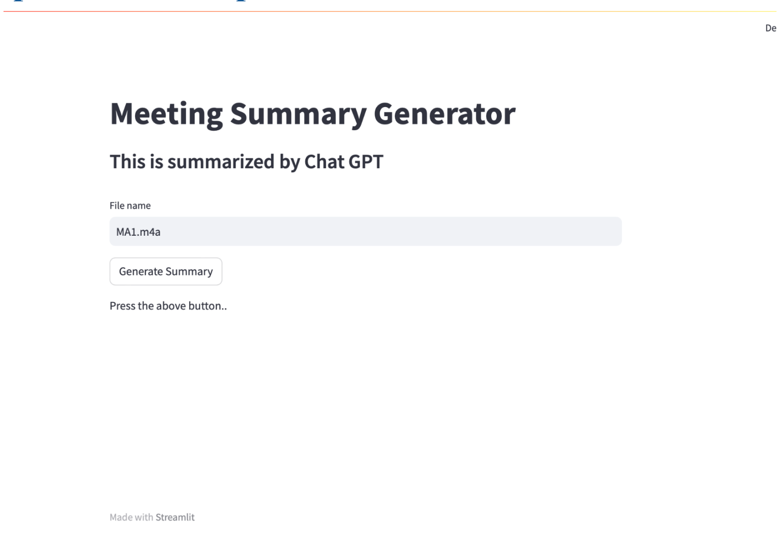
Profesor:
Juan Nolazco

Fecha de entrega:
Lunes 4 de diciembre de 2023

I. Resumen

Este reporte presenta una innovadora aplicación de inteligencia artificial para la generación de texto, específicamente, resúmenes de contenido de audio. La aplicación, que utiliza las librerías “streamlit” y “OpenAI”, permite a los usuarios introducir el nombre de un archivo de audio para su análisis y posterior resumen. Esta herramienta es especialmente útil para estudiantes que buscan resumir sus clases o profesionales que necesitan realizar minutas después de una reunión laboral. En resumen, esta aplicación promete ser una valiosa herramienta para la gestión eficiente de la información en diversas situaciones.

II. Captura de pantalla de las pruebas



Meeting Summary Generator

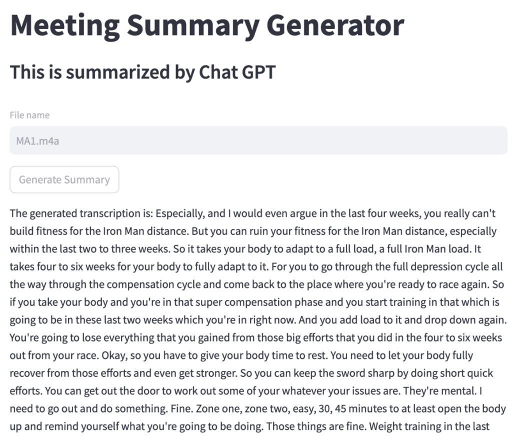
This is summarized by Chat GPT

File name

MA1.m4a

Press the above button..

Figura 1. Interfaz de Streamlit antes de usarse



Meeting Summary Generator

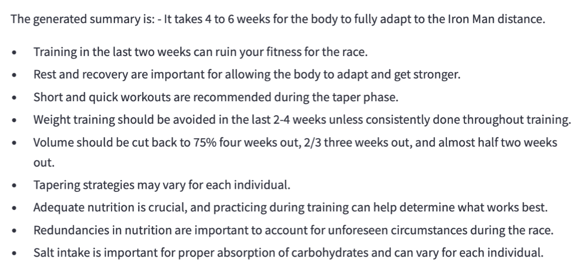
This is summarized by Chat GPT

File name

MA1.m4a

The generated transcription is: Especially, and I would even argue in the last four weeks, you really can't build fitness for the Iron Man distance. But you can ruin your fitness for the Iron Man distance, especially within the last two to three weeks. So it takes your body to adapt to a full load, a full Iron Man load. It takes four to six weeks for your body to fully adapt to it. For you to go through the full depression cycle all the way through the compensation cycle and come back to the place where you're ready to race again. So if you take your body and you're in that super compensation phase and you start training in that which is going to be in these last two weeks which you're in right now. And you add load to it and drop down again. You're going to lose everything that you gained from those big efforts that you did in the four to six weeks out from your race. Okay, so you have to give your body time to rest. You need to let your body fully recover from those efforts and even get stronger. So you can keep the sword sharp by doing short quick efforts. You can get out the door to work out some of your whatever your issues are. They're mental. I need to go out and do something. Fine. Zone one, zone two, easy, 30, 45 minutes to at least open the body up and remind yourself what you're going to be doing. Those things are fine. Weight training in the last

Figura 2. Salida del transcriptor en la interfaz de Streamlit



The generated summary is: - It takes 4 to 6 weeks for the body to fully adapt to the Iron Man distance.

- Training in the last two weeks can ruin your fitness for the race.
- Rest and recovery are important for allowing the body to adapt and get stronger.
- Short and quick workouts are recommended during the taper phase.
- Weight training should be avoided in the last 2-4 weeks unless consistently done throughout training.
- Volume should be cut back to 75% four weeks out, 2/3 three weeks out, and almost half two weeks out.
- Tapering strategies may vary for each individual.
- Adequate nutrition is crucial, and practicing during training can help determine what works best.
- Redundancies in nutrition are important to account for unforeseen circumstances during the race.
- Salt intake is important for proper absorption of carbohydrates and can vary for each individual.

Figura 3. Resumen generado en la interfaz de Streamlit

III. Anexo

```python
import os
import certifi
import ssl
import streamlit as st
import json

# Set the REQUESTS_CA_BUNDLE environment variable
os.environ['REQUESTS_CA_BUNDLE'] = certifi.where()

ssl._create_default_https_context = lambda:
ssl.create_default_context(cafile=certifi.where())

import ssl

print(ssl.get_default_verify_paths())

import requests
from requests.packages.urllib3.exceptions import InsecureRequestWarning

requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

import openai

import whisper


# Replace 'YOUR_API_KEY' with your actual OpenAI API key

openai.api_key = "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"

st.title("Meeting Summary Generator")
st.subheader("This is summarized by Chat GPT")
st.write('\n\n')

file_path = st.text_input("File name","MA1.m4a")

def load_whisper_model():

try:
```

```
model = whisper.load_model("base")

return model

except Exception as e:

print(f"Error loading Whisper model: {e}")

return None


def transcribe_audio(model, file_path):

try:

transcript = model.transcribe(file_path)

return transcript['text']

except Exception as e:

print(f"Error during transcription: {e}")

return ""
def custom_chatgpt(user_input):
messages = [{"role": "system", "content": "You are an office administrator,
summarize the text in key points"}]

messages.append({"role": "user", "content": user_input})

try:

response = openai.ChatCompletion.create(

model="gpt-3.5-turbo",

messages=messages

)

chatgpt_reply = response["choices"][0]["message"]["content"]

return chatgpt_reply
```

```
except Exception as e:

print(f"Error in ChatGPT response: {e}")

return ""
# Main Execution
model = load_whisper_model()

if model:

transcription = transcribe_audio(model, file_path)
summary = custom_chatgpt(transcription)

if st.button('Generate Summary'):
transcp_final = transcription
response_final = summary
st.write("The generated transcription is: " + transcp_final)
st.write("The generated summary is: " + response_final)
else:
st.write("Press the above button..")
```