1. Download a YouTube video's audio using yt-dlp

In [36]:
```
import yt_dlp as youtube_dl
audio_file="audio_1.mp3"
```

In [37]:
```
def get_audio(url):
    video_info = youtube_dl.YoutubeDL().extract_info(url = url, download = False)
    options = {
        'format': 'bestaudio/best',
        'keepvideo': False,
        'noplaylist': True,
        'outtmpl': audio_file,
        'postprocessors': [{
            'key': 'FFmpegExtractAudio',
            'preferredcodec': 'mp3',
            'preferredquality': '192',
        }]
    }
    with youtube_dl.YoutubeDL(options) as ydl:
        ydl.download([video_info['webpage_url']])
```

In [38]:
```
x = "https://www.youtube.com/watch?v=H-6c8GSU2d8"
get_audio(x)
```

```
[youtube] Extracting URL: https://www.youtube.com/watch?v=H-6c8GSU2d8
[youtube] H-6c8GSU2d8: Downloading webpage
[youtube] H-6c8GSU2d8: Downloading tv client config
[youtube] H-6c8GSU2d8: Downloading player c548b3da
[youtube] H-6c8GSU2d8: Downloading tv player API JSON
[youtube] H-6c8GSU2d8: Downloading ios player API JSON
[youtube] H-6c8GSU2d8: Downloading m3u8 information
[youtube] Extracting URL: https://www.youtube.com/watch?v=H-6c8GSU2d8
[youtube] H-6c8GSU2d8: Downloading webpage
[youtube] H-6c8GSU2d8: Downloading tv client config
[youtube] H-6c8GSU2d8: Downloading player c548b3da
[youtube] H-6c8GSU2d8: Downloading tv player API JSON
[youtube] H-6c8GSU2d8: Downloading ios player API JSON
[youtube] H-6c8GSU2d8: Downloading m3u8 information
[info] H-6c8GSU2d8: Downloading 1 format(s): 251
[download] audio_1.mp3 has already been downloaded
[download] 100% of    3.25MiB
[ExtractAudio] Not converting audio audio_1.mp3; file is already in target format mp
3
```

2. Transcribe the audio using Whisper.

In [39]:
```
import warnings
warnings.filterwarnings("ignore", message="FP16 is not supported on CPU; using FP32
```

In [47]:
```
import whisper
import torch
```

```
# Ensure safe loading
torch.serialization.add_safe_globals({})

def get_result():
    """Get speech recognition model."""

    device = "cuda" if torch.cuda.is_available() else "cpu"
    model = whisper.load_model("base", device=device)

    result = model.transcribe(audio_file)

    return result["text"]
```

In [48]:
```
transcript = get_result()
transcript
```

Out[48]:
```
" Elon Musk's artificial intelligence startup XAI has showed off its updated GROC-
3 model. Technology that the billionaire says is quote, the smartest AI on Earth.
Matt Blockson from Bloomberg Intelligence joins me now to assess whether or not th
at claim is true. How big, then, Matt, is this moment for the release in terms of
AI? In terms of moving things forward, has GROC done it? Has it moved the dial on
the developments and innovations around AI? Yeah, I think it's definitely moved th
e dial, but how much we'll just have to wait and see. I mean, this is like a very
early stage. It was only launched within the last hour or two and access to it is
going to be limited initially to really, plus subscribers on the X platform, which
is a $22 a month subscription. But certainly some of the claims they're making and
the demonstration that they gave earlier is pretty impressive. Given his love of S
pacey, he asked GROC to plot a course for a rocket from Earth to Mars and back aga
in. So keeping things relevant to SpaceX there. And they're claiming that it match
es and beats the other popular models like Claude and ChatGPT on a lot of complex
reasoning and calculation issues. So yeah, it's the next step forward. Yeah, and a
s you say, we'll have to see whether independent assessments verify the claim that
Musk is making, that it beats many of its competitors, talking of which, of cours
e, OpenAI is prominently amongst them. Does this put Sam Altman on the back foot t
o some extent? Is Sam Altman of OpenAI now playing catch up to Musk? Yeah, a littl
e bit, but I think this is a bit like a kind of multi-dimensional game of ping pon
g. So somebody hits them all. And then it's up to somebody else to kind of take th
ings forward. And we've seen that there recently with DeepSeat launching. And now
we have GROC 3. There's a chat GPT in the pipes. I'm sure we're going to see furth
er developments from Metro on Lama 2 and Google Gemini. So, you know, I think this
isn't the kind of the definitive statement in the world of Janice Fei and these ki
nd of chat bots. There's a lot more to come from all of Musk's competitors on thi
s, yeah, very pivotal kind of software technology."
```

3. Text Summarization: Generate a concise summary.

> summarizer = pipeline("summarization")
> summary = summarizer(text, max_length=150, min_length=50, do_sample=False)

In [42]:
```
from transformers import pipeline

device = 0 if torch.cuda.is_available() else -1
```

```
summarizer = pipeline("summarization", model="sshleifer/distilbart-cnn-12-6", revis

summary = summarizer(transcript, max_length=150, min_length=50, do_sample=False)

print(summary)
```

```
Device set to use cuda:0
[{'summary_text': " Elon Musk's artificial intelligence startup XAI has showed off i
ts updated GROC-3 model . Technology that the billionaire says is quote, the smartes
t AI on Earth . Matt Blockson from Bloomberg Intelligence joins us to assess whether
or not that claim is true ."}]
```

4. Named Entity Recognition (NER): Extract important entities (names, locations, dates, etc.).

In [43]:
```
import nltk
# nltk.download('punkt')
# nltk.download('averaged_perceptron_tagger')
# nltk.download('maxent_ne_chunker')
# nltk.download('words')
```

In [44]:
```
import spacy

# Load spaCy model for Named Entity Recognition
nlp = spacy.load("en_core_web_sm")

doc = nlp(transcript)
named_entities = [(ent.text, ent.label_) for ent in doc.ents]
print("Named Entities:\n", named_entities, "\n")
```

```
Named Entities:
 [('GROC-3', 'PERSON'), ('AI', 'GPE'), ('Earth', 'LOC'), ('Matt Blockson', 'PERSO
N'), ('Bloomberg Intelligence', 'ORG'), ('Matt', 'PERSON'), ('AI', 'GPE'), ('AI', 'G
PE'), ('the last hour', 'TIME'), ('two', 'CARDINAL'), ('22', 'MONEY'), ('Spacey', 'O
RG'), ('Earth', 'LOC'), ('Mars', 'LOC'), ('Claude', 'PERSON'), ('OpenAI', 'ORG'),
('Sam Altman', 'PERSON'), ('Sam Altman', 'PERSON'), ('OpenAI', 'GPE'), ('DeepSeat',
'ORG'), ('3', 'CARDINAL'), ('GPT', 'ORG'), ('Metro', 'FAC'), ('Lama 2', 'DATE'), ('J
anice Fei', 'PERSON'), ('Musk', 'PERSON')]
```

5. Language Detection & Translation: Detect the language and translate it into Spanish.

detected_lang = detect(text)

translated_text=GoogleTranslator(source="auto",target=target_lang).translate(text)

In [45]:
```
from langdetect import detect
from deep_translator import GoogleTranslator

def detect_and_translate(text, target_lang):
    # Detect the language of the text
    detected_lang = detect(text)
```

```
    # Translate the text to the target language
    translated_text = GoogleTranslator(source='auto', target=target_lang).translate

    print(f"Detected Language: {detected_lang}")
    print(f"Translated Text: {translated_text}")

detect_and_translate(transcript, 'es')
```

```
Detected Language: en
Translated Text: La startup de inteligencia artificial de Elon Musk, Xai, ha mostrad
o su modelo actualizado de Germ-3. La tecnología que dice el multimillonario es cit
a, la IA más inteligente de la Tierra. Matt Blockson de Bloomberg Intelligence se un
e a mí ahora para evaluar si esa afirmación es cierta o no. ¿Qué tan grande, entonce
s, Matt, es este momento para el lanzamiento en términos de IA? En términos de avanz
ar, ¿Groc lo ha hecho? ¿Ha movido el dial sobre los desarrollos e innovaciones en to
rno a la IA? Sí, creo que definitivamente ha movido el dial, pero cuánto tendremos q
ue esperar y ver. Quiero decir, esto es como una etapa muy temprana. Solo se lanzó d
entro de la última hora o dos y el acceso a él se limitará inicialmente a realmente,
además de suscriptores en la plataforma X, que es una suscripción de $ 22 por mes. P
ero ciertamente algunas de las afirmaciones que están haciendo y la demostración que
dieron anteriormente es bastante impresionante. Dado su amor por Spacey, le pidió a
Grock que planeara un curso para un cohete desde la Tierra hasta Marte y de regreso.
Así que mantener las cosas relevantes para SpaceX allí. Y afirman que coincide y sup
era a los otros modelos populares como Claude y Chatgpt en muchos problemas de razon
amiento y cálculo complejos. Así que sí, es el siguiente paso adelante. Sí, y como u
sted dice, tendremos que ver si las evaluaciones independientes verifican la afirmac
ión que Musk está haciendo, que supera a muchos de sus competidores, hablando de los
cuales, por supuesto, OpenAi está prominentemente entre ellos. ¿Esto pone a Sam Altm
an en el pie trasero hasta cierto punto? ¿Sam Altman de Operai ahora está jugando a
Musk? Sí, un poco, pero creo que esto es un poco como una especie de juego multidime
nsional de Ping Pong. Entonces alguien los golpea a todos. Y luego depende de otra p
ersona llevar adelante las cosas. Y lo hemos visto recientemente con el lanzamiento
de Deepseat. Y ahora tenemos Groc 3. Hay un chat GPT en las tuberías. Estoy seguro d
e que veremos más desarrollos de Metro sobre Lama 2 y Google Gemini. Entonces, ya sa
bes, creo que este no es el tipo de declaración definitiva en el mundo de Janice Fei
y este tipo de bots de chat. Hay mucho más por venir de todos los competidores de Mu
sk en este tipo de tecnología de software muy fundamental.
```

6. Topic Classification: Categorize the text into predefined topics (e.g., technology, politics, health, sports, etc.).

> classifier = pipeline("zero-shot-classification")

> candidate_labels = ["technology", "health", "sports", "education", "politics", "entertainment"]

> result = classifier(text, candidate_labels)

In [46]:
```
from transformers import pipeline

device = 0 if torch.cuda.is_available() else -1

# Initialize the zero-shot classification pipeline with a suitable model
classifier = pipeline("zero-shot-classification", model="facebook/bart-large-mnli",
```

```python
# Define the candidate labels
candidate_labels = ["Technology", "Health", "Sports", "Education", "Politics", "Ent

# Perform the classification
result = classifier(transcript, candidate_labels)

# Print the results
print("Topic Classification:")
for label, score in zip(result["labels"], result["scores"]):
    print(f"{label}: {score}")
```

```
Device set to use cuda:0
Topic Classification:
Technology: 0.7734097242355347
Entertainment: 0.06303556263446808
Politics: 0.05011872202157974
Sports: 0.04983419179916382
Education: 0.033371273428201675
Health: 0.030230598524212837
```