Introduction to Image Processing and Analysis with Python

This notebook provides a comprehensive guide to processing and analyzing images using Python. It leverages various libraries and tools to fetch, manipulate, and analyze images, particularly focusing on extracting and deduplicating image data, as well as integrating with cloud services for data storage and processing.

Key Libraries and Tools

1. **Python Libraries**:
   - `requests` : For fetching images from URLs.
   - `base64` : For encoding image content.
   - `pandas` : For data manipulation and analysis.
   - `PIL` (Python Imaging Library): For image processing.
   - `hashlib` : For computing image hashes.
   - `tqdm` : For progress bars.
   - `dotenv` : For loading environment variables.
   - `google.cloud.bigquery` : For interacting with Google BigQuery.
2. **LangChain**:
   - `AzureChatOpenAI` : For integrating with Azure OpenAI services.
   - `BaseModel` and `Field` from `langchain_core.pydantic_v1` : For creating structured data models.
   - `HumanMessage` and `chain` from `langchain_core.messages` and `runnables` : For creating and running message chains.
   - `JsonOutputParser` : For parsing JSON outputs.

Notebook Workflow

1. Setup and Initialization

The notebook begins by importing necessary libraries and setting up environment variables using `dotenv` . It also configures pandas display options for better readability.

In [53]:
```
import sys
from dotenv import load_dotenv
from langchain_openai import AzureChatOpenAI
import os
load_dotenv(".env", override=True)
import base64
import requests
import pandas as pd

from langchain_core.pydantic_v1 import BaseModel, Field
```

```
from langchain_core.messages import HumanMessage
from typing import Literal
from langchain_core.runnables import chain
from langchain_core.output_parsers import JsonOutputParser
from google.cloud import bigquery

from PIL import Image
from io import BytesIO
from IPython import display
import hashlib
from tqdm import tqdm
from google.cloud import storage
from PIL import Image  # Using PIL for image manipulation

from tools.octocloud import Dataframe, Table, Directory

# Pandas display options
pd.set_option('display.max_colwidth', None)
pd.set_option('display.max_colwidth', None)
```

In [54]:
```
# Constants
client = bigquery.Client(os.getenv("PROJECT_ID"))
directory = Directory(client, os.getenv("PROJECT_ID"), dataset="renault_crea")
input_ads_table = Table(client, directory, "fb_ads")
input_image_url_table = Table(client, directory, "fb_ads_image_urls")
input_video_url_table = Table(client, directory, "fb_ads_video_urls")
intermediate_image_table = Table(client, directory, "fb_ads_image_urls_intermedi
intermediate_video_table = Table(client, directory, "fb_ads_video_urls_intermedi

PAYER = "'MEDIAPLUS FRANCE', 'Peugeot'"
```

```
c:\Users\User\demo-creachecker-renault\.venv\Lib\site-packages\google\auth\_defau
lt.py:76: UserWarning: Your application has authenticated using end user credenti
als from Google Cloud SDK without a quota project. You might receive a "quota exc
eeded" or "API not enabled" error. See the following page for troubleshooting: ht
tps://cloud.google.com/docs/authentication/adc-troubleshooting/user-creds.
  warnings.warn(_CLOUD_SDK_CREDENTIALS_WARNING)
WARNING:google.auth._default:No project ID could be determined. Consider running
`gcloud config set project` or setting the GOOGLE_CLOUD_PROJECT environment varia
ble
```

2. Data import

Images

In [60]:
```
# from google.cloud import bigquery


sql_image= """
SELECT
  ad_creation_time,
  ad_delivery_stop_time,
  ad_delivery_start_time,
  ad_active_status,
  client_id,
  beneficiary,
  payer,
  id,
  image_url,
```

```
  gcs_url
FROM(
  SELECT
    *
  FROM(
    SELECT *
    FROM(
      SELECT
        ad_creation_time,
        ad_delivery_stop_time,
        ad_delivery_start_time,
        ad_active_status,
        client_id,
        id,
        image_url,
        video_url,
        beneficiary_payer.beneficiary AS beneficiary,
        beneficiary_payer.payer AS payer,
        (SELECT MAX(ad_creative_body) FROM UNNEST(ad_creative_bodies) AS ad_crea
      FROM {input_ads_table} LEFT JOIN UNNEST(video_urls) AS video_url LEFT JOIN
    )
    -- WHERE payer IN ({payer})
    -- OR payer = "OMD"
    -- only keep the rows without videos
    WHERE video_url IS NULL
  ) AS info
  LEFT JOIN
  (
    SELECT
      *EXCEPT(id)
    FROM {input_image_urls}
   ) AS image
  USING(image_url)
)
-- Only keep the rows where the gcs link is available
WHERE gcs_url IS NOT NULL
"""

query_image = sql_image.format(
    input_ads_table = input_ads_table.path("standard"),
  input_image_urls = input_image_url_table.path("standard"),
  payer = PAYER

)

print(query_image)

df_image = client.query_and_wait(query_image).to_dataframe()
print("Number of images:", len(df_image))
```

```
SELECT
  ad_creation_time,
  ad_delivery_stop_time,
  ad_delivery_start_time,
  ad_active_status,
  client_id,
  beneficiary,
  payer,
  id,
  image_url,
  gcs_url
FROM(
  SELECT
    *
  FROM(
    SELECT *
    FROM(
      SELECT
        ad_creation_time,
        ad_delivery_stop_time,
        ad_delivery_start_time,
        ad_active_status,
        client_id,
        id,
        image_url,
        video_url,
        beneficiary_payer.beneficiary AS beneficiary,
        beneficiary_payer.payer AS payer,
        (SELECT MAX(ad_creative_body) FROM UNNEST(ad_creative_bodies) AS ad_creat
ive_body) AS ad_creative_body
      FROM `ds-genai-test.renault_crea.fb_ads` LEFT JOIN UNNEST(video_urls) AS vi
deo_url LEFT JOIN UNNEST(image_urls) AS image_url, UNNEST(beneficiary_payers) AS
beneficiary_payer
    )
    -- WHERE payer IN ('MEDIAPLUS FRANCE', 'Peugeot')
    -- OR payer = "OMD"
    -- only keep the rows without videos
    WHERE video_url IS NULL
  ) AS info
  LEFT JOIN
  (
    SELECT
      *EXCEPT(id)
    FROM `ds-genai-test.renault_crea.fb_ads_image_urls`
   ) AS image
  USING(image_url)
)
-- Only keep the rows where the gcs link is available
WHERE gcs_url IS NOT NULL

Number of images: 11917
```

Video

In [61]:

```
sql_image= """
SELECT
  ad_creation_time,
  ad_delivery_stop_time,
  ad_delivery_start_time,
```

```
  ad_active_status,
  client_id,
  beneficiary,
  payer,
  id,
  video_url,
  gcs_url
FROM(
  SELECT
    *
  FROM(
    SELECT *
    FROM(
      SELECT
        ad_creation_time,
        ad_delivery_stop_time,
        ad_delivery_start_time,
        ad_active_status,
        client_id,
        id,
        image_url,
        video_url,
        beneficiary_payer.beneficiary AS beneficiary,
        beneficiary_payer.payer AS payer,
        (SELECT MAX(ad_creative_body) FROM UNNEST(ad_creative_bodies) AS ad_crea
      FROM {input_ads_table} LEFT JOIN UNNEST(video_urls) AS video_url LEFT JOIN
    )
    -- WHERE payer IN ({payer})
    -- OR payer = "OMD"
    -- only keep the rows with video
    WHERE video_url IS NOT NULL
  ) AS info
  LEFT JOIN
  (
    SELECT
      *EXCEPT(id)
    FROM {input_video_urls}
   ) AS video
  USING(video_url)
)
-- Only keep the rows where the gcs link is available
WHERE gcs_url IS NOT NULL
"""

query_video = sql_image.format(
  input_ads_table = input_ads_table.path("standard"),
  input_video_urls = input_video_url_table.path("standard"),
  payer = PAYER

)

print(query_video)

df_video = client.query_and_wait(query_video).to_dataframe()
print("Number of videos:", len(df_video))
```

```
SELECT
  ad_creation_time,
  ad_delivery_stop_time,
  ad_delivery_start_time,
  ad_active_status,
  client_id,
  beneficiary,
  payer,
  id,
  video_url,
  gcs_url
FROM(
  SELECT
    *
  FROM(
    SELECT *
    FROM(
      SELECT
        ad_creation_time,
        ad_delivery_stop_time,
        ad_delivery_start_time,
        ad_active_status,
        client_id,
        id,
        image_url,
        video_url,
        beneficiary_payer.beneficiary AS beneficiary,
        beneficiary_payer.payer AS payer,
        (SELECT MAX(ad_creative_body) FROM UNNEST(ad_creative_bodies) AS ad_creat
ive_body) AS ad_creative_body
      FROM `ds-genai-test.renault_crea.fb_ads` LEFT JOIN UNNEST(video_urls) AS vi
deo_url LEFT JOIN UNNEST(image_urls) AS image_url, UNNEST(beneficiary_payers) AS
beneficiary_payer
    )
    -- WHERE payer IN ('MEDIAPLUS FRANCE', 'Peugeot')
    -- OR payer = "OMD"
    -- only keep the rows with video
    WHERE video_url IS NOT NULL
  ) AS info
  LEFT JOIN
  (
    SELECT
      *EXCEPT(id)
    FROM `ds-genai-test.renault_crea.fb_ads_video_urls`
   ) AS video
  USING(video_url)
)
-- Only keep the rows where the gcs link is available
WHERE gcs_url IS NOT NULL

Number of videos: 4949
```

In [78]:
```
df_test_image = df_image.head().copy()
df_test_video = df_video.head().copy()
```

In [79]:
```
df_test_image
```

Out[79]:

	ad_creation_time	ad_delivery_stop_time	ad_delivery_start_time	ad_active_status	clier
0	2024-09-02	2024-10-01	2024-09-02	INACTIVE	Rer
1	2024-09-02	2024-10-01	2024-09-02	INACTIVE	Rer
2	2024-09-02	2024-10-01	2024-09-02	INACTIVE	Rer
3	2024-09-02	2024-09-30	2024-09-02	INACTIVE	Rer
4	2024-09-02	2024-09-30	2024-09-02	INACTIVE	Rer

3. Data cleaning: Image deduplication and perceptual hashing

Hashing is the transformation of any data into a usually shorter fixed-length value or key that represents the original string.

Just as we have unique Fingerprints, Hashes are unique for any particular data. There are lots of Hashing Algorithms out there which cater to specific needs.

Perceptual hash is like a regular hash in that it is a smaller, compare-able fingerprint of a much larger piece of data. However, unlike a typical hashing algorithm, the idea with a perceptual hash is that the perceptual hashes are "close" (or equal) if the original images are close.

In [65]:
```
def dhash(image, hash_size=8):
    """
    Generates a dHash for the input image.
    Args:
        image (PIL.Image.Image): Input image
        hash_size: the size to which the image has to be resized for hash calcul
    Returns:
        str: Hexadecimal representation of the hash.
```

```
    """
    resized = image.resize((hash_size + 1, hash_size), Image.LANCZOS).convert('L
    difference = []
    for row in range(hash_size):
        for col in range(hash_size):
            pixel_left = resized.getpixel((col, row))
            pixel_right = resized.getpixel((col + 1, row))
            difference.append(pixel_left > pixel_right)

    # Convert boolean list to integer, then to hex representation
    decimal_value = int("".join([str(int(b)) for b in difference]), 2)
    hash_hex = hex(decimal_value)[2:]

    # If the hex string has less digits, left pad with zeros
    hash_hex_len = hash_size * hash_size / 4 # Divide by 4 to get correct no of
    hash_hex =  hash_hex.zfill(int(hash_hex_len))

    return hash_hex
```

In [66]:

```
def image_to_bytes(gcs_url):
    """Loads an image from Google Cloud Storage and returns a PIL Image object.
    Args:
        gcs_url (str): Full GCS URL of the image (e.g., "gs://bucket/image.jpg")
    Returns:
        PIL.Image.Image: PIL Image object, or None if an error occurred.
    """
    try:
        bucket_name, blob_name = gcs_url.replace("gs://", "").split("/")[-2:]
        client = storage.Client(os.getenv("PROJECT_ID"))
        bucket = client.bucket(bucket_name)
        blob = bucket.blob(blob_name)
        image_bytes = blob.download_as_bytes()
        return image_bytes
    except Exception as e:
        print(f"Error loading image from GCS: {e}")
        return None

def image_to_pillow(gcs_url):

    image = Image.open(BytesIO(image_to_bytes(gcs_url)))
    return image


def calculate_aspect(width: int, height: int) -> str:
    def gcd(a, b):
        """The GCD (greatest common divisor) is the highest number that evenly d
        return a if b == 0 else gcd(b, a % b)

    r = gcd(width, height)
    x = int(width / r)
    y = int(height / r)

    return f"{x}:{y}"
```

In [80]:

```
for index, row in df_test_image.iterrows():
    gcs_url = row["gcs_url"] # Make sure your dataframe has gcs_url column
    image = image_to_pillow(gcs_url)
    if image:
        df_test_image.loc[index, "image_hash"] = dhash(image)
```

```
        df_test_image.loc[index, "size"] = f"{image.size[1]}:{image.size[0]}"
        df_test_image.loc[index, "aspect_ratio"] = calculate_aspect(image.size[1
    else:
        df_test_image.loc[index, "image_hash"] = None
        df_test_image.loc[index, "size"] = None
        df_test_image.loc[index, "aspect_ratio"] = None # Add None if image coul
```

```
c:\Users\User\demo-creachecker-renault\.venv\Lib\site-packages\google\auth\_defau
lt.py:76: UserWarning: Your application has authenticated using end user credenti
als from Google Cloud SDK without a quota project. You might receive a "quota exc
eeded" or "API not enabled" error. See the following page for troubleshooting: ht
tps://cloud.google.com/docs/authentication/adc-troubleshooting/user-creds.
  warnings.warn(_CLOUD_SDK_CREDENTIALS_WARNING)
WARNING:google.auth._default:No project ID could be determined. Consider running
`gcloud config set project` or setting the GOOGLE_CLOUD_PROJECT environment varia
ble
c:\Users\User\demo-creachecker-renault\.venv\Lib\site-packages\google\auth\_defau
lt.py:76: UserWarning: Your application has authenticated using end user credenti
als from Google Cloud SDK without a quota project. You might receive a "quota exc
eeded" or "API not enabled" error. See the following page for troubleshooting: ht
tps://cloud.google.com/docs/authentication/adc-troubleshooting/user-creds.
  warnings.warn(_CLOUD_SDK_CREDENTIALS_WARNING)
WARNING:google.auth._default:No project ID could be determined. Consider running
`gcloud config set project` or setting the GOOGLE_CLOUD_PROJECT environment varia
ble
c:\Users\User\demo-creachecker-renault\.venv\Lib\site-packages\google\auth\_defau
lt.py:76: UserWarning: Your application has authenticated using end user credenti
als from Google Cloud SDK without a quota project. You might receive a "quota exc
eeded" or "API not enabled" error. See the following page for troubleshooting: ht
tps://cloud.google.com/docs/authentication/adc-troubleshooting/user-creds.
  warnings.warn(_CLOUD_SDK_CREDENTIALS_WARNING)
WARNING:google.auth._default:No project ID could be determined. Consider running
`gcloud config set project` or setting the GOOGLE_CLOUD_PROJECT environment varia
ble
c:\Users\User\demo-creachecker-renault\.venv\Lib\site-packages\google\auth\_defau
lt.py:76: UserWarning: Your application has authenticated using end user credenti
als from Google Cloud SDK without a quota project. You might receive a "quota exc
eeded" or "API not enabled" error. See the following page for troubleshooting: ht
tps://cloud.google.com/docs/authentication/adc-troubleshooting/user-creds.
  warnings.warn(_CLOUD_SDK_CREDENTIALS_WARNING)
WARNING:google.auth._default:No project ID could be determined. Consider running
`gcloud config set project` or setting the GOOGLE_CLOUD_PROJECT environment varia
ble
c:\Users\User\demo-creachecker-renault\.venv\Lib\site-packages\google\auth\_defau
lt.py:76: UserWarning: Your application has authenticated using end user credenti
als from Google Cloud SDK without a quota project. You might receive a "quota exc
eeded" or "API not enabled" error. See the following page for troubleshooting: ht
tps://cloud.google.com/docs/authentication/adc-troubleshooting/user-creds.
  warnings.warn(_CLOUD_SDK_CREDENTIALS_WARNING)
WARNING:google.auth._default:No project ID could be determined. Consider running
`gcloud config set project` or setting the GOOGLE_CLOUD_PROJECT environment varia
ble
```

Export of intermediate table to BigQuery

In [72]:
```
Dataframe(client, df_test_image).to_table(intermediate_image_table)
```

```
> Exporting dataframe to table ds-genai-test.renault_crea.fb_ads_image_urls_inter
mediate
> Job 872ee49d-6c06-448e-b634-6f4c96fbda27 is currently RUNNING >>>
> Query DONE (⊙‿←)

> Email: bastien.chappis@fifty-five.com
> Job time: 2025-01-08 09:26:56.023000+00:00

Loaded 5 rows and 13 columns to ds-genai-test.renault_crea.fb_ads_image_urls_inte
rmediate
```

Out[72]:
```
LoadJob<project=ds-genai-test, location=EU, id=872ee49d-6c06-448e-b634-6f4c96fb
da27>
```

4. Feature extraction

In [ ]:
```
sql ="""
SELECT
  image_hash,
  MAX(gcs_url) AS gcs_url
FROM {intermediate_image_table}
GROUP BY 1
"""
```

In [11]:
```
query = sql.format(
    intermediate_image_table=intermediate_image_table.path("standard")
)

df = client.query_and_wait(query).to_dataframe()
print("Number of images:", len(df))
```

```
Number of images: 3
```

In [12]:
```
from google import genai
from google.genai import types

genai_client = genai.Client(
    vertexai=True, project=os.getenv("PROJECT_ID"), location='us-central1'
)
```

In [13]:
```
# Load the image from GCS
gcs_url = df["gcs_url"][0]
bytes = image_to_bytes(gcs_url)
```

```
c:\Users\User\demo-creachecker-renault\.venv\Lib\site-packages\google\auth\_defau
lt.py:76: UserWarning: Your application has authenticated using end user credenti
als from Google Cloud SDK without a quota project. You might receive a "quota exc
eeded" or "API not enabled" error. See the following page for troubleshooting: ht
tps://cloud.google.com/docs/authentication/adc-troubleshooting/user-creds.
  warnings.warn(_CLOUD_SDK_CREDENTIALS_WARNING)
```

In [81]:
```
from pydantic import BaseModel, Field, constr, field_validator
from typing import List, Union
from typing_extensions import Literal
from typing import Optional


class ImageInformation(BaseModel):
    """Information about an image."""
```

```
    image_description: str = Field(description="A brief one-sentence description
    product_type: Literal["car", "service", "connected services", "accessories"]
    contrast_presence: bool = Field(description="Is the contrast strong? True if
    product_elements: List[str] = Field(description="List of short descriptions
    product_presence: bool = Field(description="Is the car displayed in the ad?
    # product_share: int = Field(default = None, description="Percentage of vide
    product_percent: int = Field(description="Percentage of the creative area co
    main_objects: List[str] = Field(description="List of the main objects in the
    text: bool = Field(description="Indicates if there is text in the creative.
    text_location: Literal["top", "down", "right", "left", "center", "None"]= Fi
    brand_logo: bool = Field(description="Indicates if there is a brand logo in
    promotion: bool = Field(description="Indicates if there is a promotion call
    promotion_deadline: bool = Field(description="Indicates if a promotion deadl
    promotion_theme: str= Field(default= "None", description="Specifies if the p
    call_to_action: bool = Field(description="Indicates if there is a call to ac
    call_to_action_size: int = Field(default=None, description="Percentage of th
    car_brand: str = Field(default="None", description="The brand of the car, if
    car_model: str = Field(default="None", description="The model of the car, if
    price: bool = Field(description="Indicates if a price is displayed in the cr
    ad_purpose: str = Field(default="None", description="The intended action pro
    text_extraction: str = Field(default="None", description="The text content e
    price_size: int = Field(default="None", description="Percentage of the creat
    color_palette: List[str] = Field(description="Dominant colors used in the cr
    human_elements: List[str] = Field(default="None", description="Short descrip
    human_presence: bool = Field(description="Are there humans displayed in the
    human_number: int = Field(default="None", description="Number of humans in t
    human_genders: List[str] = Field(default="None", description="List of humans
    imagery_type: Literal["photographs", "illustration", "mix media", "infograph
    style: List[str] = Field(description="General aesthetic and style of the cre
    actions: List[str] = Field(description="Activities or actions taking place i
    actions_presence: bool = Field(description="Is there an action/story taking
    environment: Literal["indoor", "outdoor", "urban", "rural"] = Field(descript
    time_of_day: Literal["not identifiable", "daytime", "nighttime", "sunset", "
    weather: Literal["None", "sunny", "cloudy", "rainy"] = Field(default="None",
    background_main_color: str = Field(default="None", description="Dominant col
    human_shot_size: List[str] = Field(default="None", description="Shot size of
    human_age: Literal["senior", "adult", "child", "baby"] = Field(default="None
    electric_vehicle: bool = Field(description="Is the ad related to the electri
    goal_awareness_or_consideration_or_conversion: str = Field(default="None", d
    # logo_display: Optional[bool] = Field(None, description="Is the logo displa
    # branding_share: Optional[int] = Field(None, description="Percentage of the
    black_bars: bool = Field(description="Is there a black bar on the creative?
    text_length: int = Field(default = None, description="Number of lines the te
    promotion_display: bool = Field(description="Indicates whether a promotion m
    call_to_action_verb: str = Field(default="None", description="The specific v
    promotion_wording_type: Literal["None", "gain", "loss_aversion"] = Field(def
    promotion_tone: Literal["None", "informative", "persuasive", "emotional", "h
    new_old_vehicle: Literal["None", "new", "second-hand"] = Field(default="None
    # # a_audio_presence: Optional[bool] = Field(None, description="Is there aud
    # # a_subtitles_presence: Optional[bool] = Field(None, description="Are ther
    product_usage: bool = Field(description="Is the product (car, connected feat
    # # d_action_audio: Optional[bool] = Field(None, description="Is there a cle

response = genai_client.models.generate_content(
    model='gemini-2.0-flash-exp',
    contents=[
        'Describe the content of the picture',
        types.Part.from_bytes(data=bytes, mime_type="image/png")
    ],
    config=types.GenerateContentConfig(
```

```
        temperature=0,
        response_mime_type= 'application/json',
        response_schema= ImageInformation
    )
)
print(response.text)
```


```
{
  "actions": [],
  "actions_presence": false,
  "black_bars": false,
  "brand_logo": true,
  "call_to_action": false,
  "color_palette": ["gray", "red", "black"],
  "contrast_presence": true,
  "electric_vehicle": false,
  "environment": "outdoor",
  "human_presence": false,
  "image_description": "A gray Mini Countryman with a red roof is shown from the
rear, parked on a red surface with a selection of color and wheel options display
ed below.",
  "imagery_type": "3D rendering",
  "main_objects": ["car"],
  "new_old_vehicle": "new",
  "price": false,
  "product_elements": ["car", "color options", "wheel options"],
  "product_percent": 80,
  "product_presence": true,
  "product_type": "car",
  "product_usage": false,
  "promotion": false,
  "promotion_deadline": false,
  "promotion_display": false,
  "style": ["modern", "clean"],
  "text": true,
  "time_of_day": "daytime",
  "ad_purpose": "None",
  "background_main_color": "white",
  "call_to_action_size": 0,
  "call_to_action_verb": "None",
  "car_brand": "Mini",
  "car_model": "Countryman",
  "goal_awareness_or_consideration_or_conversion": "consideration",
  "human_age": "adult",
  "human_elements": [],
  "human_genders": [],
  "human_number": 0,
  "human_shot_size": [],
  "price_size": 0,
  "promotion_theme": "None",
  "promotion_tone": "None",
  "promotion_wording_type": "None",
  "text_extraction": "MINI COUNTRYMAN",
  "text_length": 1,
  "text_location": "center",
  "weather": "sunny"
}
```


In [42]:
```
image_to_pillow(gcs_url)
```

```
c:\Users\User\demo-creachecker-renault\.venv\Lib\site-packages\google\auth\_defau
lt.py:76: UserWarning: Your application has authenticated using end user credenti
als from Google Cloud SDK without a quota project. You might receive a "quota exc
eeded" or "API not enabled" error. See the following page for troubleshooting: ht
tps://cloud.google.com/docs/authentication/adc-troubleshooting/user-creds.
  warnings.warn(_CLOUD_SDK_CREDENTIALS_WARNING)
WARNING:google.auth._default:No project ID could be determined. Consider running
`gcloud config set project` or setting the GOOGLE_CLOUD_PROJECT environment varia
ble
```

Out[42]:

Estimate costs

Princing list

Princing list 2

In [45]:

```python
# input_tokens = genai_client.models.count_tokens(
#     model='gemini-2.0-flash-exp',
#     contents=[
#         'Hello?',
#         types.Part.from_bytes(data=bytes, mime_type="image/png")
#     ],
#     config=types.GenerateContentConfig(
#         temperature=0,
```

```
#         response_mime_type= 'application/json',
#         response_schema= ImageInformation
#     )
# )

input_tokens = response.usage_metadata.prompt_token_count
output_tokens = response.usage_metadata.candidates_token_count
intermediate_tokens = response.usage_metadata.cached_content_token_count

if intermediate_tokens:
    print("Cached tokens: ", intermediate_tokens)

print("Input tokens: ", input_tokens)
print("Output tokens:", output_tokens)

NUMBER_IMAGE = len(df)
price = input_tokens/1E3*0.00001875 + output_tokens/1E3*0.000075
image_price = 0.00002
print("Price $:", (price + image_price) * NUMBER_IMAGE)
```

```
Input tokens:  1312
Output tokens: 409
Price $: 1.0320202499999998
```

In [ ]:

5. Video extraction

In [73]:
```
df_test_video
```

Out[73]:

	ad_creation_time	ad_delivery_stop_time	ad_delivery_start_time	ad_active_status	clier
0	2024-10-02	2024-10-10	2024-10-02	INACTIVE	Rei G
1	2024-10-02	2024-10-10	2024-10-02	INACTIVE	Rei G
2	2024-10-02	2024-10-11	2024-10-02	INACTIVE	Rei G
3	2024-10-02	2024-10-11	2024-10-02	INACTIVE	Rei G
4	2024-10-02	2024-10-11	2024-10-02	INACTIVE	Rei G

In [ ]:

```python
from pydantic import BaseModel, Field, constr, field_validator
from typing import List, Union
from typing_extensions import Literal
from typing import Optional


class VideoInformation(BaseModel):
    """Information about an image."""

    image_description: str = Field(description="A brief one-sentence description
    product_type: Literal["car", "service", "connected services", "accessories"]
    contrast_presence: bool = Field(description="Is the contrast strong? True if
    product_elements: List[str] = Field(description="List of short descriptions
    product_presence: bool = Field(description="Is the car displayed in the ad?
    product_share: int = Field(default = None, description="Percentage of video
    product_percent: int = Field(description="Percentage of the creative area co
    main_objects: List[str] = Field(description="List of the main objects in the
    text: bool = Field(description="Indicates if there is text in the creative.
    text_location: Literal["top", "down", "right", "left", "center", "None"]= Fi
    brand_logo: bool = Field(description="Indicates if there is a brand logo in
    promotion: bool = Field(description="Indicates if there is a promotion call
    promotion_deadline: bool = Field(description="Indicates if a promotion deadl
    promotion_theme: str= Field(default= "None", description="Specifies if the p
    call_to_action: bool = Field(description="Indicates if there is a call to ac
    call_to_action_size: int = Field(default=None, description="Percentage of th
    car_brand: str = Field(default="None", description="The brand of the car, if
    car_model: str = Field(default="None", description="The model of the car, if
    price: bool = Field(description="Indicates if a price is displayed in the cr
    ad_purpose: str = Field(default="None", description="The intended action pro
    text_extraction: str = Field(default="None", description="The text content e
    price_size: int = Field(default="None", description="Percentage of the creat
    color_palette: List[str] = Field(description="Dominant colors used in the cr
    human_elements: List[str] = Field(default="None", description="Short descrip
    human_presence: bool = Field(description="Are there humans displayed in the
    human_number: int = Field(default="None", description="Number of humans in t
    human_genders: List[str] = Field(default="None", description="List of humans
    imagery_type: Literal["photographs", "illustration", "mix media", "infograph
    style: List[str] = Field(description="General aesthetic and style of the cre
    actions: List[str] = Field(description="Activities or actions taking place i
    actions_presence: bool = Field(description="Is there an action/story taking
    environment: Literal["indoor", "outdoor", "urban", "rural"] = Field(descript
    time_of_day: Literal["not identifiable", "daytime", "nighttime", "sunset", "
    weather: Literal["None", "sunny", "cloudy", "rainy"] = Field(default="None",
    background_main_color: str = Field(default="None", description="Dominant col
    human_shot_size: List[str] = Field(default="None", description="Shot size of
    human_age: Literal["senior", "adult", "child", "baby"] = Field(default="None
    electric_vehicle: bool = Field(description="Is the ad related to the electri
    goal_awareness_or_consideration_or_conversion: str = Field(default="None", d
    logo_display: bool = Field(description="Is the brand logo displayed in the f
    branding_share: int = Field(None, description="Percentage of the creative wh
    black_bars: bool = Field(description="Is there a black bar on the creative?
    text_length: int = Field(default = None, description="Number of lines the te
    promotion_display: bool = Field(description="Indicates whether a promotion m
    call_to_action_verb: str = Field(default="None", description="The specific v
    promotion_wording_type: Literal["None", "gain", "loss_aversion"] = Field(def
    promotion_tone: Literal["None", "informative", "persuasive", "emotional", "h
    new_old_vehicle: Literal["None", "new", "second-hand"] = Field(description="
    audio_presence: bool = Field(default="None", description="Is there audio (mu
    a_subtitles_presence: bool = Field(None, description="Are there subtitles in
```

```
    product_usage: bool = Field(description="Is the product (car, connected feat
    d_action_audio: bool = Field(description="Is there a clear audio call to act
```

In [82]:

```
video_uri = df_test_video["gcs_url"][0]

# Display the Video
display(Video(video_uri))

video_analysis_prompt = """You are an expert video analyser, for each 10 second

contents = [
    types.Part.from_uri(
        file_uri=video_uri,
        mime_type="video/*",
    ),
    video_analysis_prompt,
]

response = genai_client.models.generate_content(
    model='gemini-2.0-flash-exp',
    contents=contents,
    config=types.GenerateContentConfig(
        temperature=0,
        response_mime_type= 'application/json',
        response_schema= ImageInformation
    )
)
print(response.text)
```


0:00 / 3:00

```
{
  "actions": ["talking"],
  "actions_presence": true,
  "black_bars": false,
  "brand_logo": true,
  "call_to_action": false,
  "color_palette": ["yellow", "blue", "white"],
  "contrast_presence": true,
  "electric_vehicle": true,
  "environment": "indoor",
  "human_presence": true,
  "image_description": "A woman in a blue suit is talking to the camera.",
  "imagery_type": "photographs",
  "main_objects": ["woman", "sofa"],
  "new_old_vehicle": "new",
  "price": false,
  "product_elements": ["car"],
  "product_percent": 10,
  "product_presence": true,
  "product_type": "car",
  "product_usage": false,
  "promotion": false,
  "promotion_deadline": false,
  "promotion_display": false,
  "style": ["modern"],
  "text": true,
  "time_of_day": "daytime",
  "ad_purpose": "inform",
  "background_main_color": "yellow",
  "call_to_action_size": 0,
  "call_to_action_verb": "None",
  "car_brand": "Renault",
  "car_model": "Renault 5 E-Tech electric",
  "goal_awareness_or_consideration_or_conversion": "awareness",
  "human_age": "adult",
  "human_elements": ["woman with long blonde hair"],
  "human_genders": ["female"],
  "human_number": 1,
  "human_shot_size": ["medium shot"],
  "price_size": 0,
  "promotion_theme": "None",
  "promotion_tone": "informative",
  "promotion_wording_type": "None",
  "text_extraction": "Oh. I received a message. Aurélien planet Cléa, it's your t
urn! Okay, right on the topic. Let's go. A FUTURGEN story by Cléa Martinet VP Sus
tainability, Renault Group & Ampere Ep.4/4: Renault 5 E-Tech electric: The ultima
te answer to decarbonization challenges 3 words to describe you? So, I guess, I'd
say enthusiastic Deeply to what I do And, I'm a go getter. 10 seconds to explain
the EV societal issues? So I guess I'd sum it up by um saying that um being an al
l electric car maker we have to navigate through three paradoxes. The first one b
eing we have to be electric and being affordable. Secondly is being electric whil
e preserving the resources. And third one is being electric while being responsib
le all along the value chain. How do we do it? So what we do to navigate those th
ree paradoxes is that we have a life cycle approach from the conception to the en
d of life. So we tackle each step of the way in the most economic, competitive, l
ow carbon way possible. So we source low carbon materials, we assemble the car in
France in our industrial plants in northern France and we conceive a car that is
lightweight and really economical and sober in the use of resources and particula
rly electricity. Tell us more? So what this very concretely and what this vehicle
does is embodying the ESG native component of all that. The car is assembled in n
```

orthern France in plants that are going to be carbon neutral in 2025. The battery will come from literally 300m from our industrial plants from 2025 and on. The logistics and suppliers ecosystem is really virtuous. 75% of them are located within a 300 kilometer radius. And the sourcing of the vehicle is really virtuous too in terms of circular economy, up to 26.5% of materials for the vehicle come from the circular economy and the vehicle in itself is 85% recyclable. Any last words? This car really makes me very happy. Because there's something a bit, you know, quite magic about this R5 which went through the ages of the past century with its, you know, very joyful, revolutionary, witty, smart way of tackling crisis. And it's quite a happy thing to see it being modern, still modern today and being revisited in an all electric version. So I really do hope it will go through the ages once again and make new generations as happy as they made us in the past. Sorry, I'm not supposed to parler comme ça, ça va? Je me suis tourné vers le mec du son, excuse-moi. Non encore tes tempans? Pardon. FUTURGEN stories",
  "text_location": "down",
  "weather": "None"
}