

# Hackathon

## Pipeline Ingestion – Analitiche Sentiment Analysis – Pipeline per l’Uso di Generative AI

Ilaria Picariello  
Claudio Spasiano

May 2025

## 1 Scraping manuale da Instagram con Selenium e GUI

Lo script `interactive_gui_comment_scraper.py` consente di estrarre manualmente i commenti da un post Instagram pubblico, attraverso un’interfaccia grafica interattiva. Questo approccio è necessario poiché Instagram non fornisce API pubbliche affidabili per accedere ai commenti, ed impone limitazioni dinamiche basate su autenticazione, comportamento utente e struttura HTML.

### Utilizzo di Selenium

**Selenium** è una libreria di automazione per browser che consente di simulare l’interazione umana con un sito web (click, scroll, lettura DOM). Viene utilizzato per:

- avviare il browser Chrome con un profilo personalizzato,
- accedere a `instagram.com` come utente autenticato,
- permettere all’utente di navigare manualmente verso un post specifico,
- ispezionare dinamicamente la struttura HTML della pagina per estrarre i commenti.

L’interazione con la pagina è resa affidabile tramite identificatori XPath personalizzati che selezionano blocchi di commenti e metadati associati.

### Autenticazione tramite cookie

Instagram richiede l’accesso autenticato per visualizzare o interagire con i commenti. Poiché l’autenticazione interattiva manuale non è sostenibile in contesti ripetuti, viene utilizzata una tecnica di **autenticazione tramite cookie**:

- L'utente effettua il login una sola volta (tramite uno script separato o manualmente),
- I cookie di sessione vengono salvati localmente nel file `instagram_cookies_scraper.json`,
- All'avvio, lo script carica e inietta i cookie nel browser tramite Selenium,
- Dopo il refresh della pagina, l'utente risulta automaticamente autenticato.

Ciò consente di mantenere un accesso persistente e sicuro, evitando ripetuti login via UI e riducendo il rischio di blocchi da parte di Instagram.

### **Estrazione dei commenti**

Una volta caricata la pagina di un post pubblico:

1. L'utente clicca sul pulsante “Estrai Commenti e Salva” nella GUI,
2. Lo script legge la struttura HTML della pagina con Selenium,
3. Identifica blocchi di commenti contenuti in `<ul><div>...</div></ul>`,
4. Per ogni blocco:
  - Estrae l'username dell'autore,
  - Isola il testo del commento ignorando elementi come “Mi piace”, “Rispondi”, timestamp o contatori,
  - Conta i like se disponibili,
  - Estrae emoji se presenti.
5. I commenti sono convertiti in dizionari JSON coerenti con lo schema del dataset principale.

I commenti vengono poi salvati in `data/gui_extracted_comments.json`.

### **Interfaccia grafica (GUI)**

L'interfaccia è realizzata con **Tkinter**, libreria standard di Python per GUI. Consente di:

- visualizzare un messaggio di istruzioni,
- lanciare lo scraping al click su un pulsante,
- mostrare i risultati in tempo reale,
- notificare l'avvenuto salvataggio con una finestra popup.

## 2 Refresh token su Reddit

Lo script di scraping per Reddit (`reddit_praw_ingestion.py`) utilizza la libreria ufficiale PRAW (Python Reddit API Wrapper) per accedere ai commenti.

Reddit applica una politica di sicurezza OAuth 2.0, che richiede:

- un `client_id` e `client_secret`,
- un access token temporaneo (scade ogni ora),
- la generazione di un **refresh token** per rigenerare in automatico nuovi access token.






A tal fine, è disponibile uno script ausiliario `reddit_login.py` che guida l'utente nella generazione manuale del **refresh token**, permettendo così una raccolta di commenti Reddit più stabile, sicura e senza login interattivi ricorrenti.

## 3 Pipeline TikTok

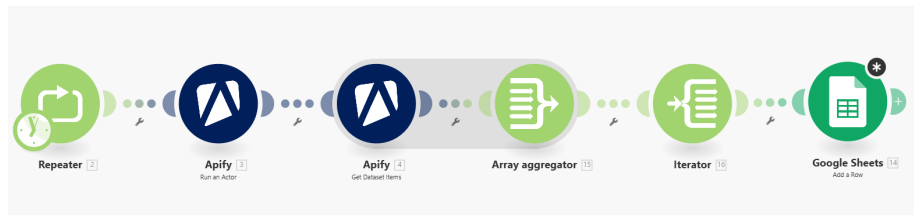
Lo scraping di TikTok è avvenuto in maniera differente rispetto agli altri social media. Si è infatti, tramite lo store, Apify, deciso di usare l'actor "TikTok Data extractor". Tramite ciò è stato possibile effettuare scraping con un buon grado di personalizzazione, ci si è infatti potuti concentrare sui trend più comuni riguardanti il GP di Monaco, oppure si è potuto selezionare il profilo da cui fare scraping e l'arco temporale come illustrato qui di seguito

The screenshot displays the configuration interface for the 'TikTok Data extractor' actor. It is divided into two main sections. The first section, titled 'Videos with this hashtag (optional)', contains a list of four input fields for hashtags: '#MonacoGp', '#Verstappen', '#F1', and '#Leclerc'. Each field has a red 'X' icon to its right. Below these fields are two buttons: '+ Add' and 'Bulk edit', and a 'Remove empty fields' button on the right. The second section, titled 'Number of videos per hashtag, profile or search (optional)', features a numeric input field set to '50' with '+' and '-' buttons, and a dropdown menu currently showing 'videos'. Below this are two expandable sections: 'Profiles' (indicated by a person icon) and 'Search' (indicated by a magnifying glass icon).

Ottenendo quindi un output del genere

#	Author's Avatar authorMeta.avatar	Author authorMeta.name	Text text	Diggs diggCount	Shares shareCount	Plays playCount	Comments commentCount	Duration (seconds) videoMeta.duration	Music musicMeta.musicName	Music author musicMeta.musicAuthor
1		f1	MONACO BABYYYY 🏆 #f1 #formula1 #monacogp...	80800	1897	505500	555	35	original sound	Formula 1
2		mundodeportivo	🇪🇸 @Fabio Marchi lo tiene claro: "El GP de Mónaco ha sido un...	14800	255	425900	350	136	sonido original	Mundo Deportivo
3		parismatch	#charlesleclerc #tiktoksports #monacogp...	141	2	3267	2	17	original sound	MELODY
4		dazn_es	🏎️ Carlos Sainz, muy duro contra lo ocurrido en Mónac...	20800	651	411300	222	95	sonido original - DAZN España	DAZN España
5		baw6teen	ouch #f1 #f2 #monacogp #monaco #fyp	64000	11200	703600	87	11	original sound	🌟 Nostalgia vibe 🌟

Al fine di filtrare tramite una pipeline automatica tutti i dati ottenuti si è utilizzato il tool Make.com

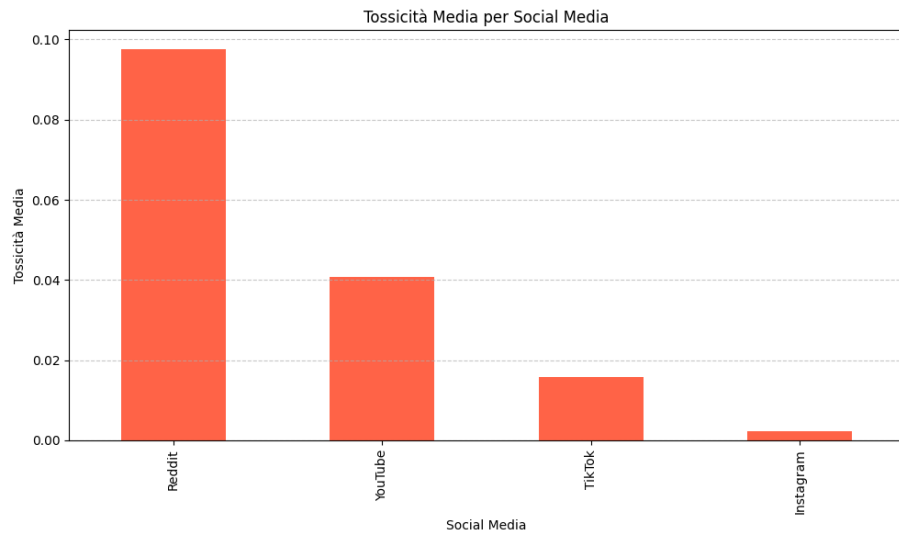


Nelle prime fasi si crea una connessione con quanto fatto da Apify facendo un importazione dei dati ottenuti, tutti i dati vengono organizzati in forma tabellare e poi canalizzati fino all'ultimo modulo in cui si è deciso di mantenere solo un certo numero di colonne in modo da attenersi al dataset di riferimento. Il tool è stato molto utile tuttavia a causa delle limitazioni imposte dal piano gratuito che forniva un credito limitato non è stato possibile massimizzare le opportunità da esso offerte sia in termini di personalizzazione sia in termini numerici.

## 4 Analitiche

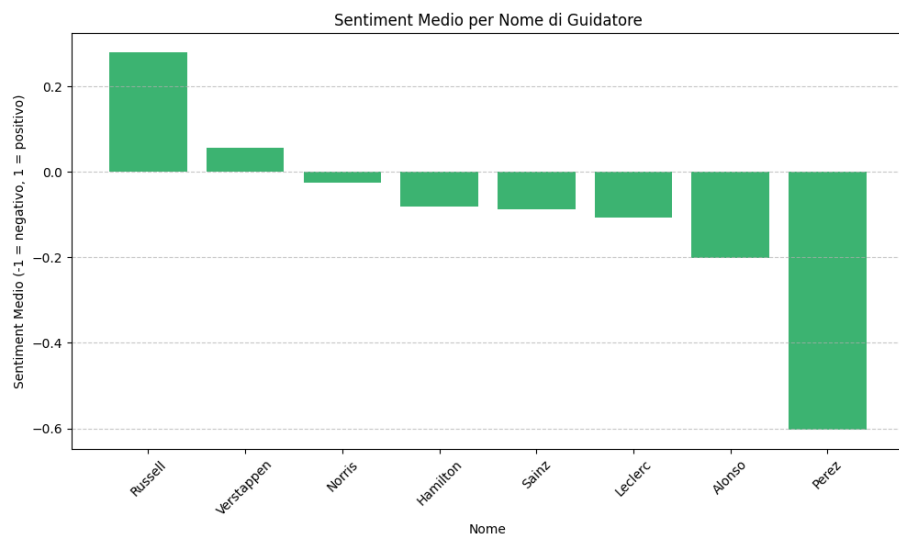
### 4.1 Tossicità media per social

Tramite toxic-bert si è assegnato un punteggio di toxicity ad ogni istanza del dataset, successivamente si è analizzato il punteggio di toxicity medio per ogni social ottenendo i seguenti risultati.



## 4.2 Sentiment medio per nome e per fase

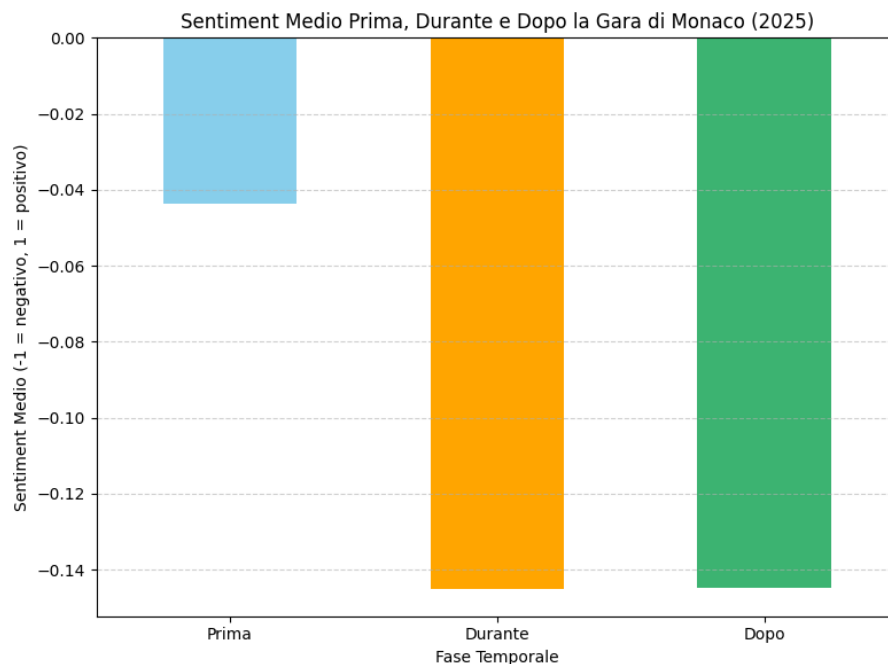
Tramite bert-multilingual si è assegnato un valore categorico (positivo, negativo, neutro) ad ogni istanza del dataset, e successivamente si è analizzato il sentiment medio per una cerchia ristretta di piloti, ottenendo i seguenti risultati



E' bene notare come Perez, nonostante non sia pilota in gara, bensì un pilota presente fino a poco fa con ReBull è comunque oggetto di molti commenti. Si rileva inoltre come Alonso, uscito presto dalla gare a causa di un problema,

riflette effettivamente un sentiment negativo. Va comunque ricordato che il dataset su cui è stata effettuata l'analisi contiene molte istanze precedenti alla gara stessa, e ciò giustifica un sentiment positivo per Russel in quanto quarto in classifica generale nonostante sia arrivato 11esimo a Monaco.

E' stato inoltre analizzato il sentiment anche su tre periodi temporali diversi ottenendo i seguenti risultati



A causa della natura temporale stessa dell'ingestion si nota, come preven-  
tivabile, che non c'è molta differenza tra il durante e dopo la gara. Generalmente  
prima della gara si registrano dei sentimenti meno sbilanciati verso il negativo  
e più neutri mentre il sentimento negativo si canalizza immediatamente dopo la  
gara

### 4.3 Time Series sentiment

Tramite lo script "time\_series\_sentiment" si analizza l'andamento temporale del  
sentiment positivo (calcolato con RoBERTa) per diversi cluster tematici. In  
particolare quello che viene fatto è salvare l'andamento giornaliero del sentiment  
positivo per ciascun cluster ed in uscita, come verrà osservato successivamente,  
si ottiene un grafico per ciascun cluster.

Proprio i cluster sono ottenuti tramite lo script "clustering.py". In questo  
script si effettua un cluster tematico automatico su un insieme di test e si assegna  
un'etichetta descrittiva per ciascun cluster. In particolare vengono mantenuti

solo i testi in lingua inglese e con testo pulito, i testi vengono trasformati in una matrice numerica con la rappresentazione TF-IDF e viene infine applicato Kmeans con  $k=6$ . Fatto ciò per ogni cluster in base ai precedenti pesi TF-IDF vengono estratte le parole più rappresentative e questo vengono trasformate in etichette leggibili come nomi dei cluster.

A valle dell'intero processo si sono ottenuti i seguenti risultati:

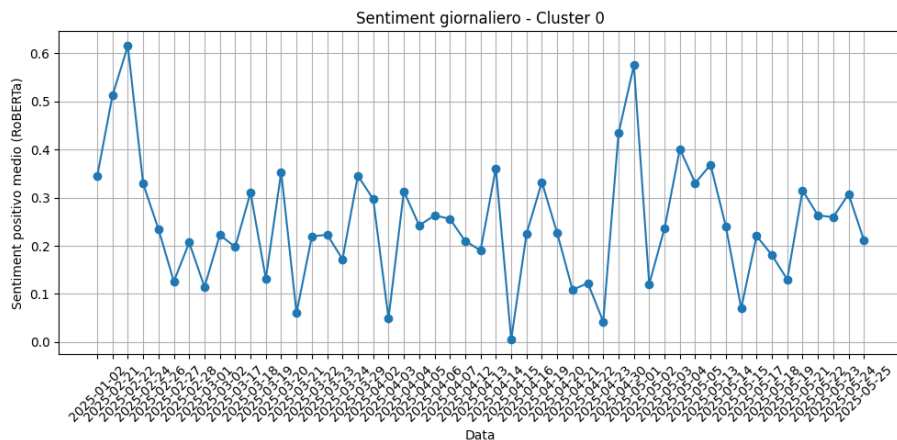


Figure 1: Cluster basato su car, like, monaco, look, safeti, get, year, track, lap, driver

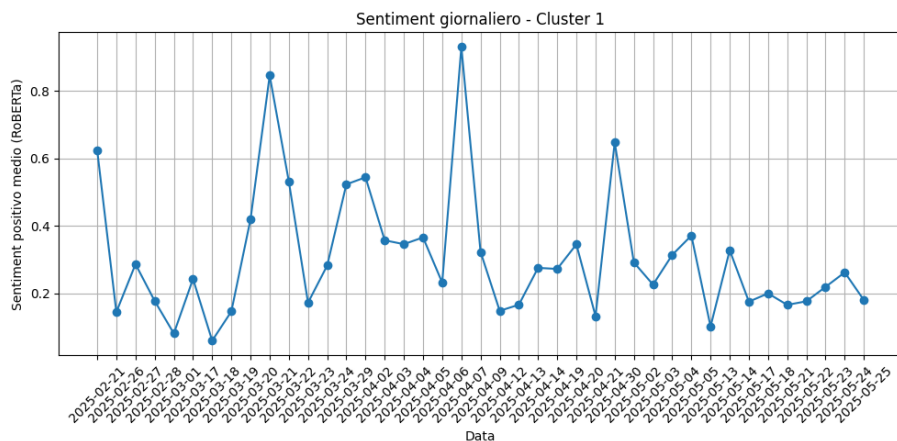


Figure 2: Cluster basato su charl, time, lap, ferrari, lewi, get, go, first, monaco, like”,

Si noti come intorno alla data del 17 marzo si verifica un picco negativo,

indice della cattiva prestazione della ferrari e di Charles Leclerc nel GP in Australia

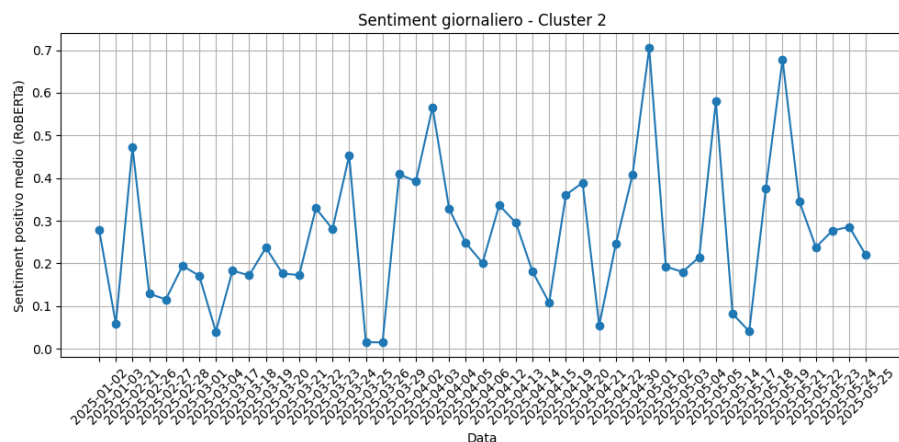


Figure 3: Cluster basato su race, monaco, bore, one, watch, get, f1, year, like, make”

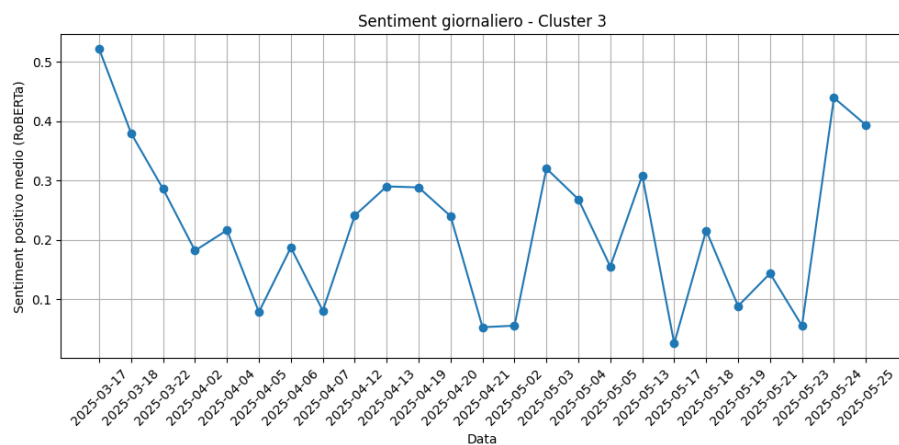


Figure 4: Cluster basato su lando, oscar, lap, max, get, mclaren, win, like, charl, race”,

Si noti il picco positivo intorno 17 Marzo in quanto Lando Norris ha vinto la gara nel GP diMelborune tenutosi il 16, Stesse considerazioni valgono per il picco intorno al 24 Maggio nel GP di Monaco sempre vinto da Lando Norris



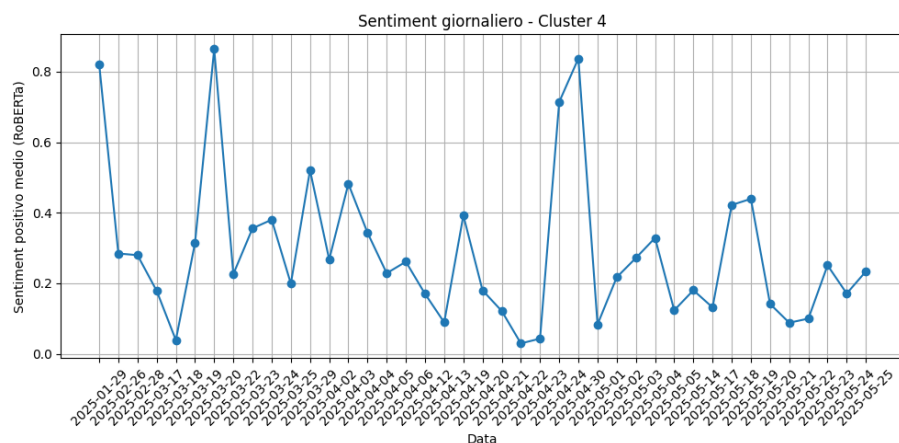


Figure 5: Cluster basatp su max, yuki, oscar, lap, get, lewi, mclaren, race, fuck, like”,

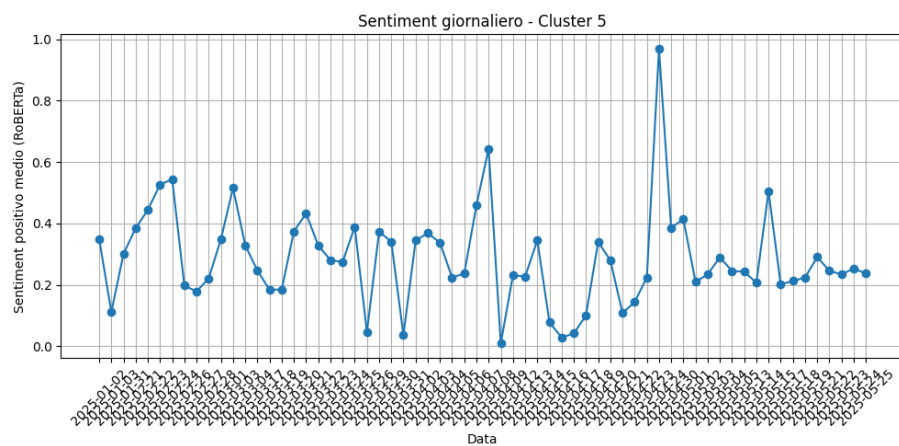


Figure 6: Cluster basato su ferrari, get, fuck, one, lol, that, year, driver, go, lap

Questo sentiment ha una componente , come si evince dal cluster, più ironica o sarcastica che ne può spiegare l’instabilità

## 5 Pipeline di Generazione Contestuale con Falcon-7B-Instruct

### 5.1 Obiettivo

Questa pipeline è progettata per utilizzare un modello generativo di linguaggio naturale, nello specifico **Falcon-7B-Instruct**, al fine di generare automaticamente descrizioni testuali per ciascun cluster tematico identificato nei commenti social relativi al Gran Premio di Monaco 2025.

Ogni cluster è stato generato tramite clustering su base TF-IDF e arricchito con:

- parole chiave salienti,
- sentiment medio,
- andamento temporale del sentiment (prima, durante e dopo la gara),
- entità nominate più rilevanti (es. piloti, team),
- engagement (like, commenti, repost).

Il modello Falcon-7B viene dunque utilizzato non per rispondere liberamente a prompt arbitrari, ma per generare **descrizioni sintetiche e contestuali** che interpretano in linguaggio naturale le caratteristiche di ciascun cluster.

L'impiego di un Large Language Model (LLM) in questa fase ha l'obiettivo di:

- trasformare insight numerici e tecnici in testi leggibili e comunicabili anche a stakeholder non tecnici (es. giornalisti, analisti, partner),
- supportare la generazione automatica di reportistica narrativa su dati social complessi,
- standardizzare e semplificare il lavoro di interpretazione dei risultati del clustering tematico,
- offrire uno strumento adattabile a casi futuri in cui si voglia riapplicare il framework con dati differenti (es. altri eventi sportivi o politici).

### 5.2 Struttura della pipeline

#### 5.2.1 Installazione delle dipendenze

Il notebook installa le principali librerie necessarie:

- **transformers** – per il caricamento e la gestione dei modelli Hugging Face,
- **accelerate** – per l'ottimizzazione dell'esecuzione su GPU o ambienti multi-dispositivo,

- `bitsandbytes` – per il supporto a modelli quantizzati (più leggeri in memoria).

```
!pip install transformers accelerate bitsandbytes --quiet
```

### 5.2.2 Autenticazione Hugging Face

Per scaricare il modello da Hugging Face Hub, viene richiesto un token di accesso personale:

```
from huggingface_hub import login
login("YOUR-SECRET-TOKEN")
```

### 5.2.3 Caricamento del modello

Il modello utilizzato è `tiiuae/falcon-7b-instruct`, un LLM da 7 miliardi di parametri ottimizzato per seguire istruzioni in linguaggio naturale.

```
from transformers import AutoTokenizer, AutoModelForCausalLM

model_id = "tiiuae/falcon-7b-instruct"
tokenizer = AutoTokenizer.from_pretrained(model_id)
model = AutoModelForCausalLM.from_pretrained(
    model_id,
    device_map="auto",
    torch_dtype="auto"
)
```

### 5.2.4 Creazione della pipeline generativa

Il modello è configurato tramite `pipeline()` per il task di generazione testuale:

```
from transformers import pipeline

generator = pipeline(
    "text-generation",
    model=model,
    tokenizer=tokenizer,
    max_new_tokens=512,
    do_sample=True,
    temperature=0.7
)
```

I parametri scelti bilanciano controllo e creatività nella generazione.

### 5.3 Prompting controllato con LangChain

Per generare descrizioni coerenti e contestuali per ciascun cluster tematico, la pipeline utilizza **LangChain**, una libreria pensata per orchestrare l'interazione con modelli linguistici attraverso prompt modulari e dati strutturati.

LangChain è stato impiegato per costruire un *prompt template parametrico* che combina più fonti di dati (keyword, sentiment, entità, andamento) in un'unica istruzione testuale fornita al modello.

#### Prompt template

Il prompt utilizzato è il seguente:

```
cluster_prompt = PromptTemplate.from_template(
    """You are a social media analyst for the Monaco GP 2025.

    Topic: {topic_label}

    Recent sentiment trend:
    {daily_scores}

    Most mentioned entities (by total likes):
    {entity_likes}

    Write 2-3 sentences summarizing how the sentiment evolved over time.
    Then, comment on which drivers or teams got the most attention.
    Do not invent events or motivations.
    Use only the data provided above."""
)
```

#### Contenuto dei parametri del prompt

- **topic\_label**: descrizione sintetica del tema del cluster, generata a partire da parole chiave TF-IDF.
- **daily\_scores**: andamento temporale del sentiment positivo calcolato con RoBERTa (`avg_roberta_pos`) per ciascun giorno.
- **entity\_likes**: elenco delle entità di tipo **PERSON** e **ORG** più menzionate nel cluster, ordinate per numero totale di like ricevuti.

#### Obiettivo del prompt

Il modello riceve come input una scheda sintetica per ciascun cluster, contenente:

- l'evoluzione reale del sentiment nel tempo,
- i soggetti (piloti, team) che hanno ricevuto maggiore attenzione,

- il contesto tematico emerso dal clustering semantico.

A partire da questi elementi, il modello genera una descrizione strutturata in 2–3 frasi, con le seguenti caratteristiche:

- evita speculazioni o eventi non menzionati nei dati ( “*Do not invent events or motivations*”),
- sintetizza trend osservati nel tempo,
- commenta le dinamiche di attenzione attorno a soggetti specifici,
- traduce dati strutturati in narrazione testuale leggibile.

#### **Esempio di output generato**

“The sentiment in this cluster started neutral and declined sharply on race day, reflecting disappointment. Ferrari and Leclerc received the most attention, often in critical tones.”

### **5.4 Ruolo di LangChain**

LangChain è stato fondamentale per:

- strutturare prompt riutilizzabili e parametrizzati;
- integrare in modo coerente più fonti di dati (trend, entità, label);
- rendere la generazione di descrizioni automatica, scalabile e riproducibile su nuovi cluster.

L’approccio modulare consente di sostituire facilmente il modello LLM, adattare il prompt a nuove analisi, o tradurre le uscite in altre lingue. La pipeline è quindi facilmente estendibile anche ad altri eventi, domini o dataset sociali.

### **5.5 Conclusioni**

Questa pipeline integra il potenziale descrittivo degli LLM con un approccio strutturato di analisi dati. Automatizza la generazione di testi interpretativi su cluster tematici di dati social, facilitando il lavoro di analisi qualitativa, sintesi e reportistica. È facilmente adattabile ad altri contesti con dati testuali etichettati e arricchiti semanticamente.