WEBRANKING

Prova pratica Innovation Engineer

- Progettazione di una pipeline di dati resiliente che riceva dati da Google Sheet e da Google Analytics, automatizzi l'estrazione giornaliera e svolga analisi sui dati per fornire insight significativi
- Si prediliga il linguaggio Python e l'ambiente Cloud quanto possibile
- Si descriva la progettazione dell'architettura, inclusi eventuali diagrammi per chiarire le fasi operative e il processo decisionale

Alberto Mastromarino

Componenti architetturali principali

Data Source principale

Linguaggio e libreria utilizzati per estrarre, manipolare e pulire i dati

Archivio online per i dati estratti e ripuliti |Data warehouse per l'analisi dei dati con query SQL

Libreria utilizzata per la parte di data visualization

Strumento di orchestrazione per l'automazione dei processi ETL, programmando i job e monitorando le esecuzioni.

Fasi del processo

Apache Airflow

Automation

Data Source → **Extract** → **Transform** → **Load**

Google Analytics

Google APIs

python

python

pandas

↓

Analysis

Matplotlib

Potenziali punti di failure e strategie di mitigazione

API non disponibile: Implementati retry automatici e gestione delle eccezioni in Python per garantire la continuità del servizio.

Errori di caricamento su BigQuery: Validazione dei dati e gestione delle autorizzazioni prima del caricamento. Logging dettagliato per il monitoraggio.

Superamento delle quote di GCS/BigQuery: Monitoraggio dell'uso delle risorse con alert per evitare interruzioni.

Soluzioni di storage preferite e motivazioni

Google Cloud Storage (GCS): Utilizzato per archiviare file CSV e JSON, garantendo sicurezza e facilità di gestione.

- **Motivazione**: Bassi costi, elevata disponibilità e integrazione con BigQuery.

- **BigQuery: Data warehouse per analisi su larga scala.**
  - **Motivazione:** Supporto per query complesse e aggregazioni su dati di grandi dimensioni con prestazioni elevate.

Modalità di Automazione della Pipeline

- **Automazione giornaliera:** La pipeline estrae i dati automaticamente ogni giorno tramite script Python e Google Sheets API.
- **Integrazione con BigQuery:** I dati vengono caricati e aggregati per analisi rapide e dettagliate.
- **Supporto per dati storici:** La pipeline gestisce l'estrazione di dati passati, permettendo di rielaborare dataset per periodi specifici
- **Tecnologia utilizzata:** Scheduler tramite Airflow per gestire i processi in modo automatizzato.

Gestione degli Errori

- **Gestione delle eccezioni in Python:** Uso di blocchi try-except per errori di connessione alle API e problemi durante la manipolazione dei dati.
- **Retry automatici:** Tentativi ripetuti in caso di errori temporanei, come indisponibilità delle API di Google Sheets o problemi di caricamento su BigQuery.
- **Errori gestiti:** Origine dati non disponibile, dataset vuoti, modifiche nelle API e problemi di autorizzazione su BigQuery.

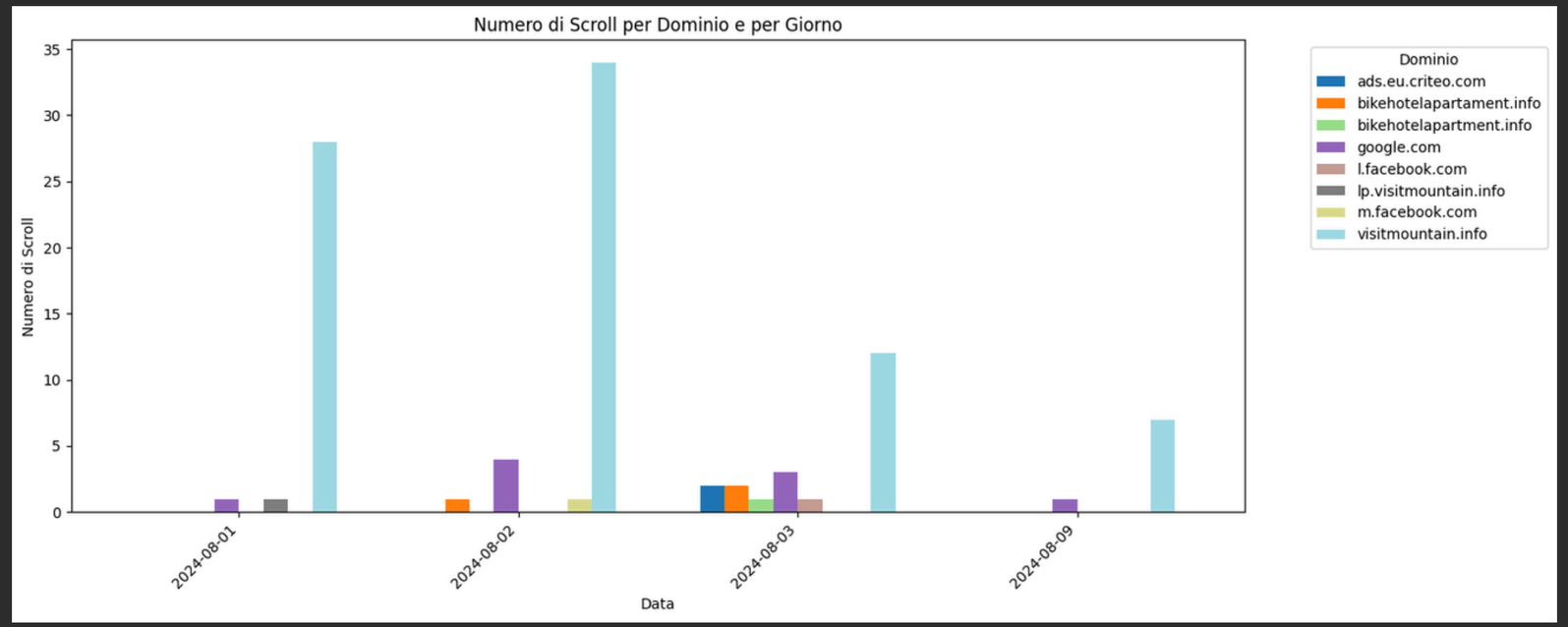
Codice

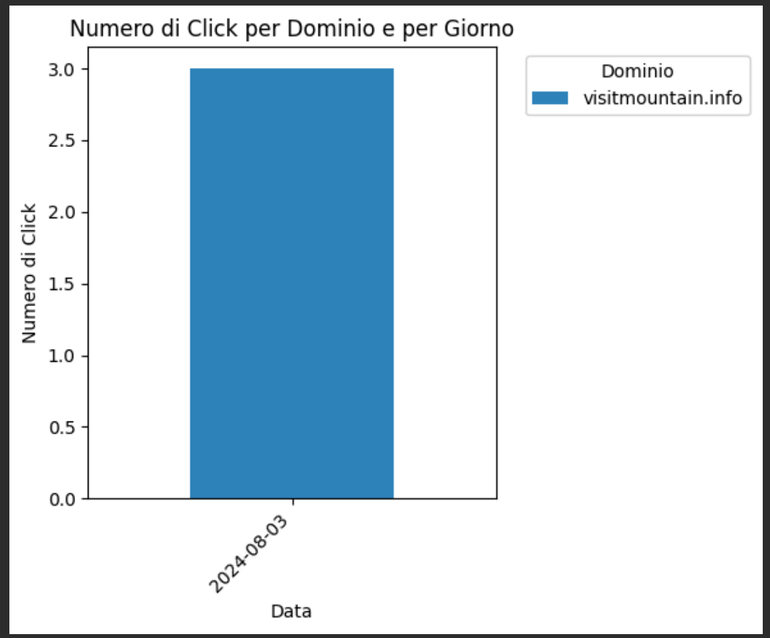
Il codice utilizzato per questo lavoro è possibile trovarlo all'interno di questo link: https://github.com/AlbeMastro/Innovation-Engineer.git

Snippet di codice per leggere i dati da Google Sheet

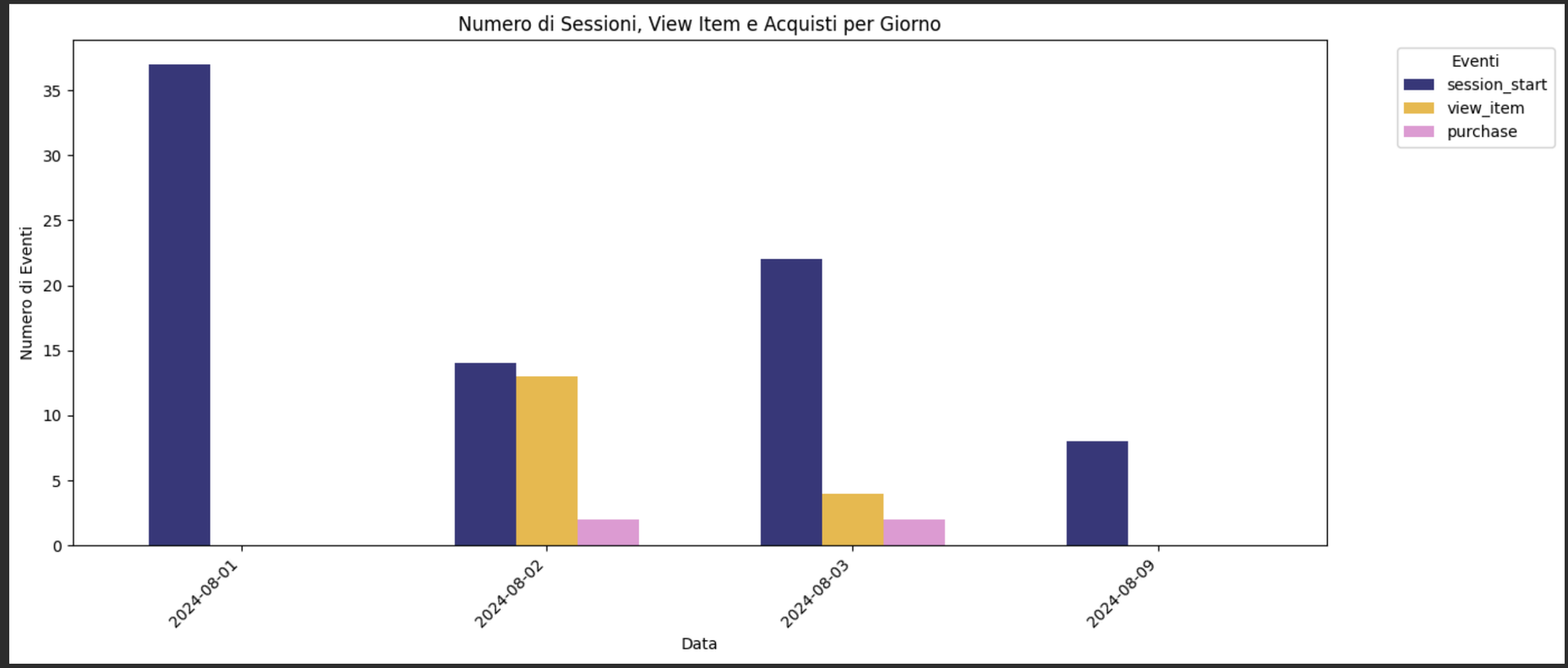
```python
# Configurazione delle credenziali
creds_path = '/Users/Alberto/PycharmProjects/Pipeline/dags/innovation-engineer-2.json'
scopes = ['https://www.googleapis.com/auth/spreadsheets']

try:
    creds = Credentials.from_service_account_file(creds_path, scopes=scopes)
    client = gspread.authorize(creds)
    print("Autenticazione Google Sheets riuscita.")
except Exception as e:
    print(f"Errore durante la configurazione delle credenziali: {e}")
    return

# Estrazione dei dati dai Google Sheets con meccanismo di retry
max_retries = 3
for attempt in range(max_retries):
    try:
        sheet_id_budget = '1412wCpCV0TQKaPHuWBM5_Ujca4f9_H8lqCJRyfoWcE0'
        sheet_budget = client.open_by_key(sheet_id_budget)
        worksheet_budget = sheet_budget.worksheet('Forecasts')

        sheet_id_ga = '18VtAi1StMUfblg4NahC5eyoFw8oCWYmLckk8F_kZ6vY'
        sheet_ga = client.open_by_key(sheet_id_ga)
        worksheet_ga = sheet_ga.worksheet('GA')

        # Conversione dei dati in DataFrame
        data_budget = worksheet_budget.get_all_records()
        df_budget = pd.DataFrame(data_budget)

        data_ga = worksheet_ga.get_all_records()
        df_ga = pd.DataFrame(data_ga)
```

Data Visualisation

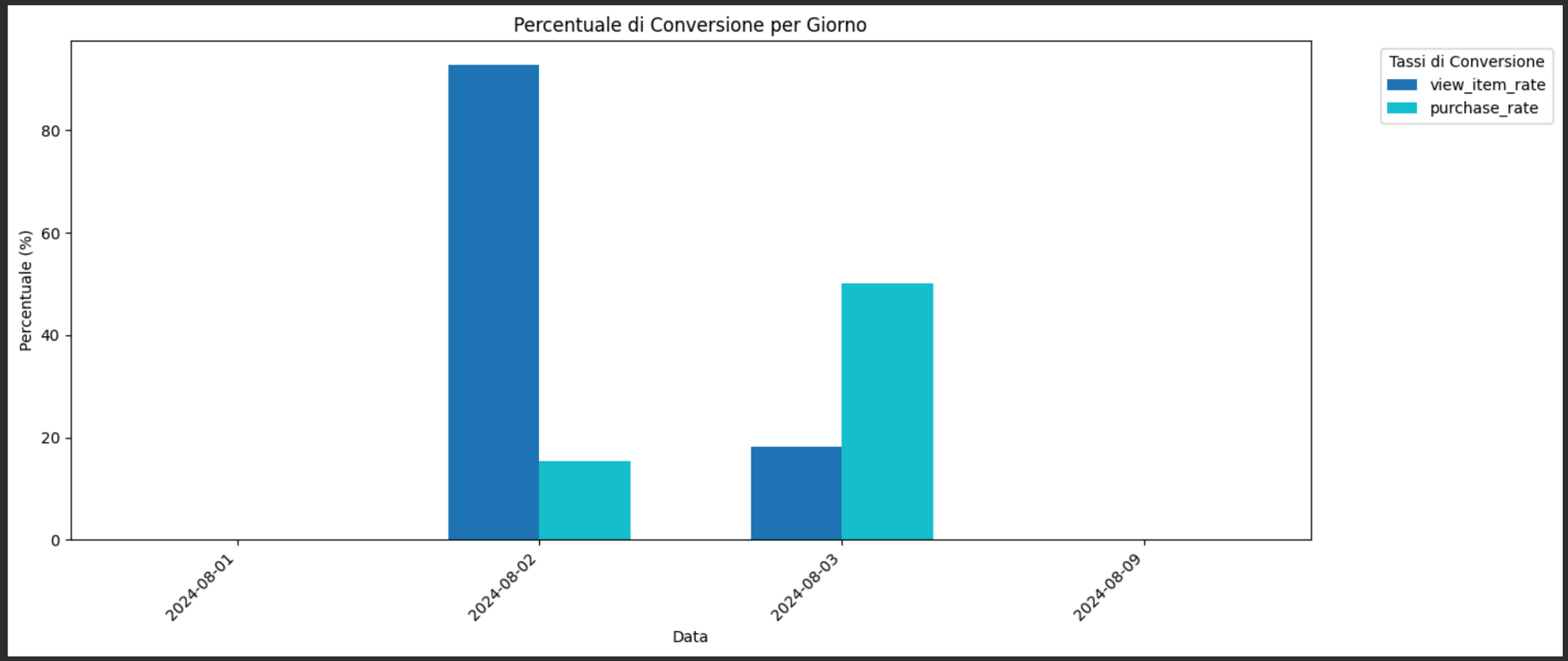




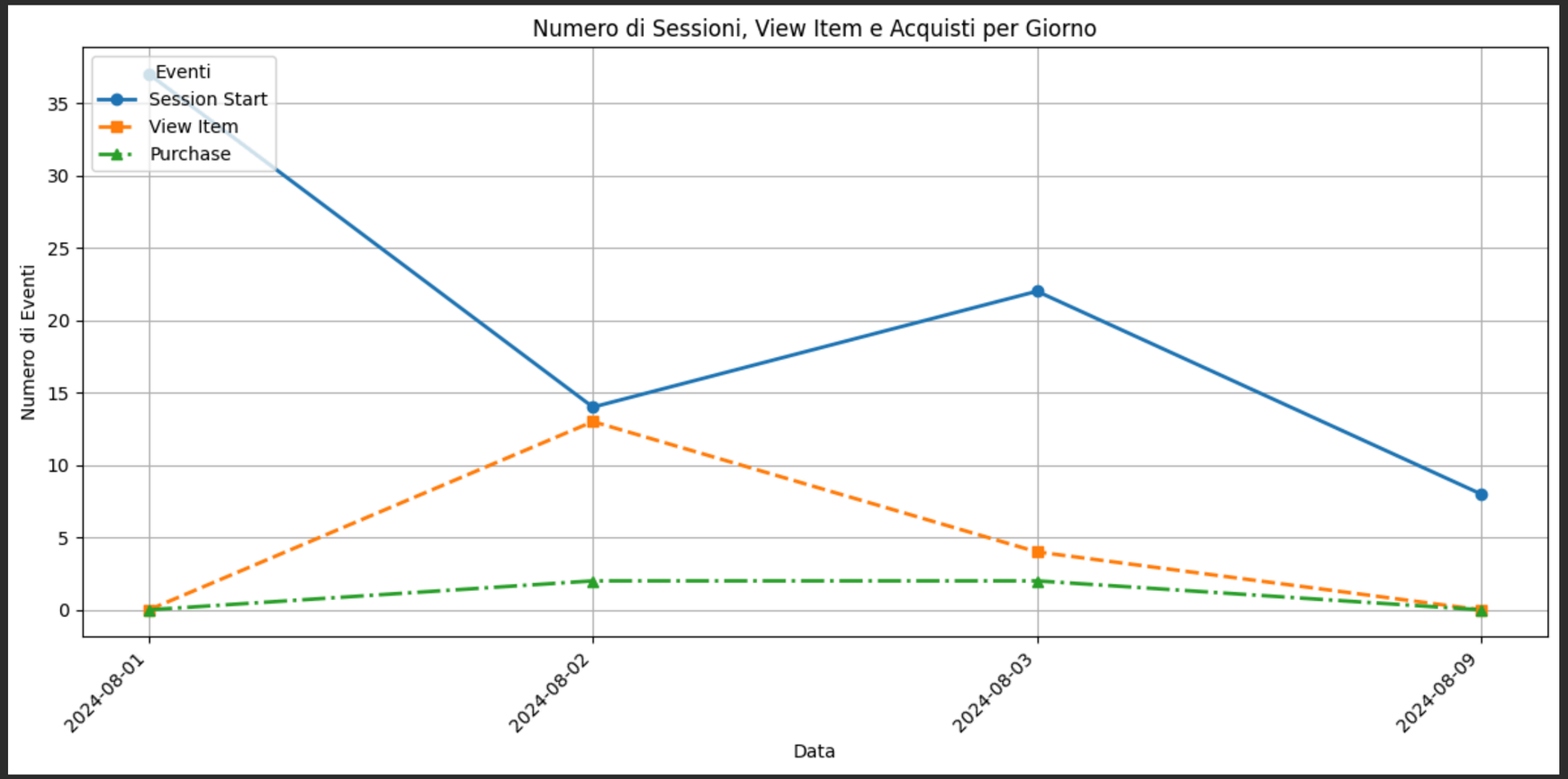
Data Visualisation



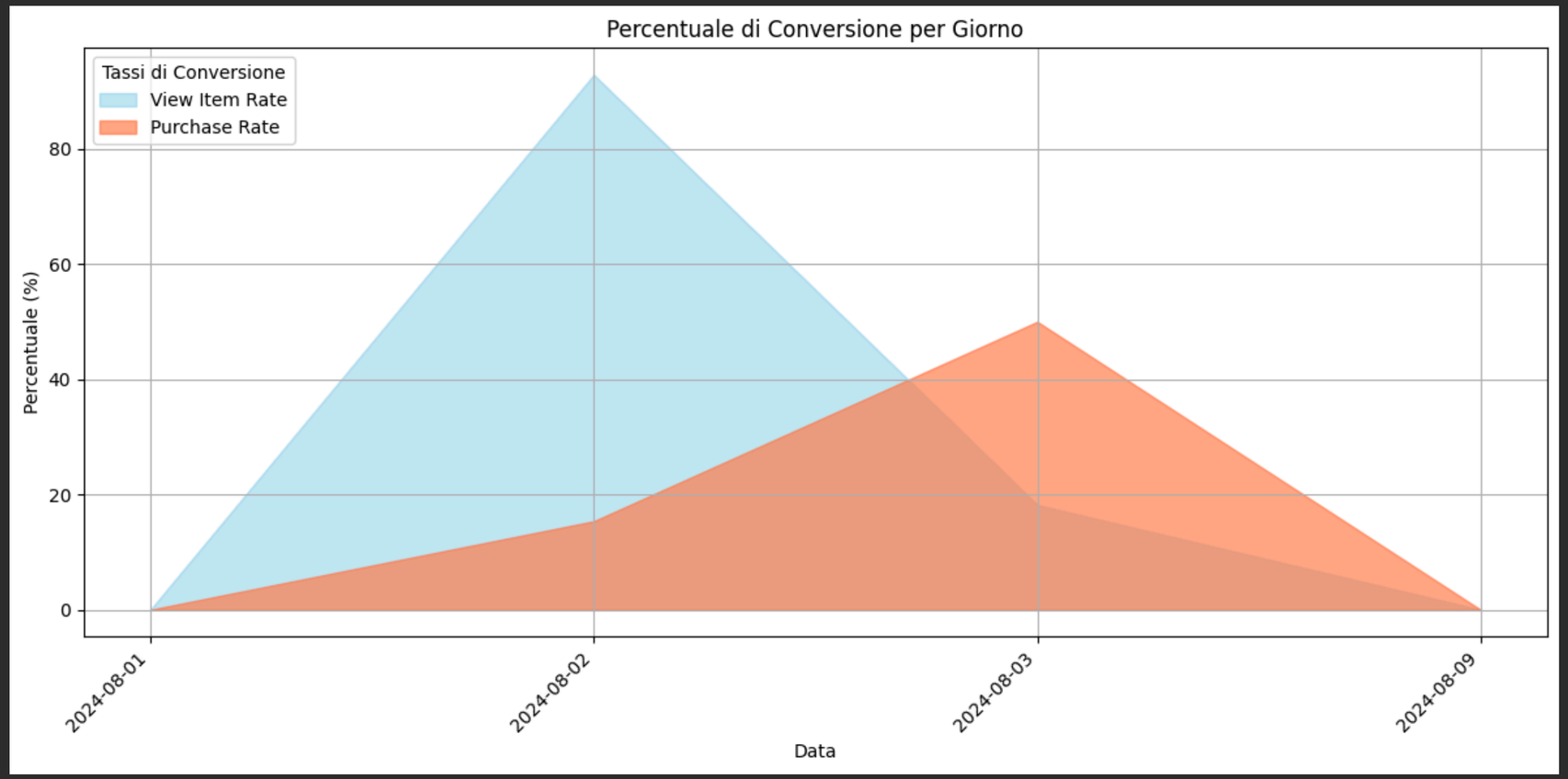
Data Visualisation



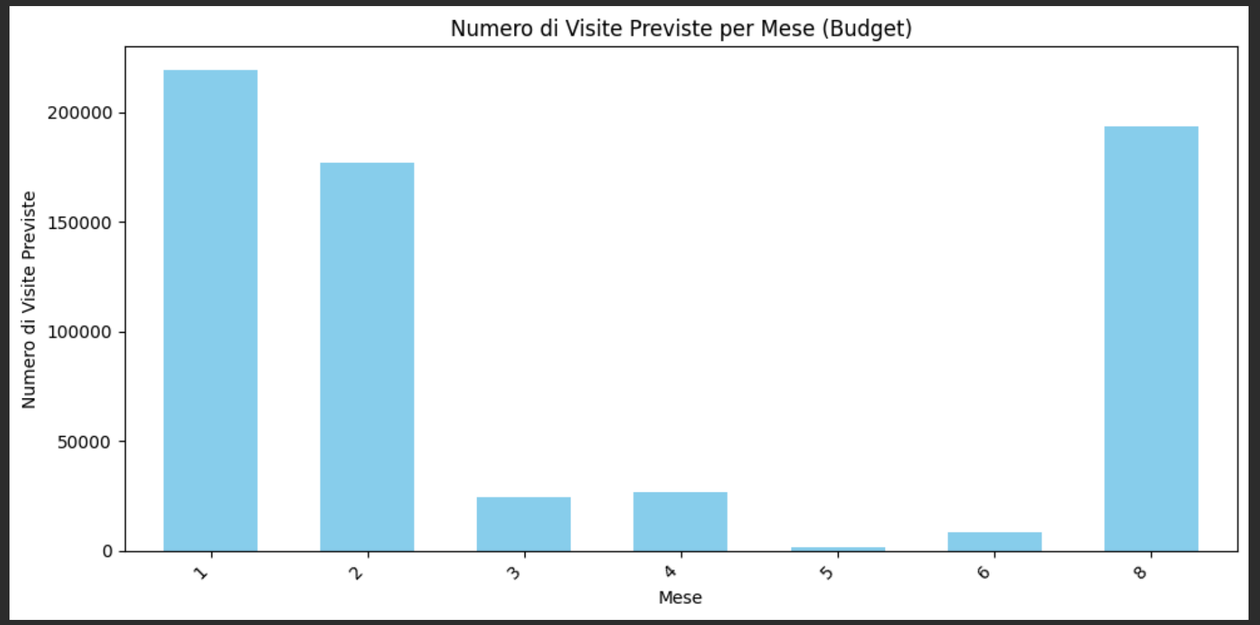
Data Visualisation

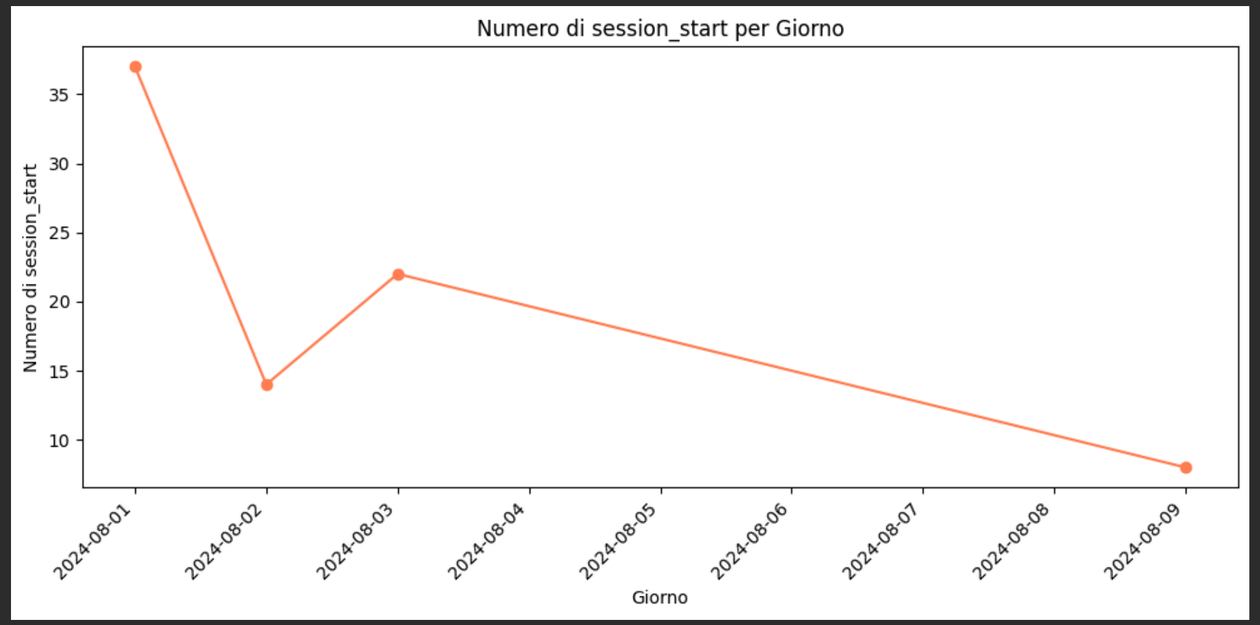


Data Visualisation



Data Visualisation





Query SQL

```sql
-- Analisi dei Click

SELECT
  DATE(Date) AS day,
  event_param_string_value AS Dominio,
  SUM(CASE WHEN event_name = 'click' THEN 1 ELSE 0 END) AS Numero_Click,
  SUM(CASE WHEN event_name = 'scroll' THEN 1 ELSE 0 END) AS Numero_Scroll
FROM
  `innovation-engineer.marketing_data.ga_data`
WHERE
  event_name IN ('click', 'scroll')
  --AND event_param_key IN ('page_location', 'page_referrer')
GROUP BY
  day, Dominio
ORDER BY
  day, Dominio;
```

```sql
-- Analisi metriche di performance

WITH funnel_data AS (
  SELECT
    DATE(Date) AS day,
    COUNTIF(event_name = 'session_start') AS session_count,
    COUNTIF(event_name = 'view_item') AS view_item_count,
    COUNTIF(event_name = 'purchase') AS purchase_count
  FROM
    `innovation-engineer.marketing_data.ga_data`
  WHERE
    event_name IN ('session_start', 'view_item', 'purchase')
  GROUP BY
    day
)
SELECT
  day,
  session_count,
  view_item_count,
  purchase_count,
  SAFE_DIVIDE(view_item_count, session_count) * 100 AS view_item_rate,
  SAFE_DIVIDE(purchase_count, view_item_count) * 100 AS purchase_rate,
  SAFE_DIVIDE(purchase_count, session_count) * 100 AS overall_conversion_rate
FROM
  funnel_data
ORDER BY
  day;
```

Query SQL

```sql
-- Analisi sessioni previste vs sessioni reali

SELECT
  month,
  SUM(`Forecasted Visits`) AS Visite_Previste
FROM
  `innovation-engineer.marketing_data.budget_data`
WHERE
  channel = 'Search'
  AND country = 'IT'
GROUP BY
  month
ORDER BY
  month;


SELECT
  COUNT(CASE WHEN event_name = 'session_start' AND source = 'google' AND medium = 'cpc' THEN 1 END) AS VisiteReali
FROM
  `innovation-engineer.marketing_data.ga_data`
WHERE event_param_key IN ('page_location','page_referrer')
```

Grazie per l’attenzione