TrEnD

Samuele Ferri, Steven Gotti, Simone Sudati

2 - AWS Glue PySpark

Data cleaning

+ Nell'analizzare i dataset forniti, è risultato chiaro che ci fossero dei problemi e dei dati mancanti

+ All'interno dei dataset abbiamo rimossi i **duplicati** e gli eventuali record con **ID non validi** presenti

+ Il problema degli ID non validi è causato dalla presenza di caratteri **escape '\n'** nella descrizione di alcuni video e ciò porta alla creazione di nuove righe non valide nel file .csv

02

Data cleaning

Abbiamo stampato in output il risultato del cleaning dei dati

OUTPUT:

```
>TEDX DATASET NULL IDX count (PRE) 4494
>TEDX DATASET NULL IDX count (PRE) 4467

>TEDX DATASET DUPLICATE count (PRE) 4467
>TEDX DATASET DUPLICATE count (POST) 4467

>TAGS DATASET DUPLICATE count (PRE) 42566
>TAGS DATASET DUPLICATE count (POST) 42566

>WATCH NEXT DATASET DUPLICATE count (PRE) 77364
>WATCH NEXT DATASET DUPLICATE count (POST) 30254
```

-27 ID non validi rimossi

Nel dataset *tedx_dataset* erano presenti record non validi

-47 110 duplicati rimossi

Nel dataset *watch_next_dataset* erano invece presenti molti duplicati

03

```
tedx_dataset = tedx_dataset.filter(length("idx") == 32)
tedx_dataset = tedx_dataset.dropDuplicates()
```

Data cleaning

Siamo riusciti ad eliminare i record con ID non validi che corrispondevano alle linee del .csv che iniziavano non con un ID ma con la continuazione della descrizione del video precedente.

Tuttavia i primi campi del video incriminato vengono ugualmente salvati nella collezione senza i campi successivi alla descrizione.

```
15  d564abb00bb146ff75c7ed2d9b48aadc,Elisabeth Pierre,Et si la bière était féminine ? | Elisabeth Pierre | TEDxToulouse,"Cette présentation
    des conférences TED.
16
17  Elisabeth Pierre est passionnée des bières et connaît toutes les facettes de sa boisson préférée. Elle aime aussi casser les stéréotype
    masculine.
18
19  Elisabeth est zythologue, une des quelques femmes experte en bière. Son travail et ses recherches avec des producteurs, chercheurs et c
    développement en font une référence sur le sujet. Une grande part de son message insiste sur le lien fort entre la bière et les femmes
    2020,https://www.ted.com/talks/elisabeth_pierre_l_histoire_inedite_des_femmes_a_l_origine_de_la_biere,0
```

La prima riga rimane salvata

Le altre nate dall'escape '\n' vengono rimosse per via ID non validi

FILTER {'_id':'d564abb00bb146ff75c7ed2d9b48aadc'}

Filtrato sul video incriminato

QUERY RESULTS 1-20 OF MANY

Esempio corretto

```
_id: "08438d7fc43df207af7e95180f6599e9"
main_speaker: "Péter Fankhauser"
title: "Meet Rezero, the dancing ballbot"
details: "Engineering student Péter Fankhauser demonstrates Rezero, a robot that..."
posted: "Posted Nov 2011"
url: "https://www.ted.com/talks/peter_fankhauser_meet_rezero_the_dancing_bal..."
> tags: Array
count_wn: 3
> watch_next_s: Array
```

QUERY RESULTS 1-1 OF 1

```
_id: "d564abb00bb146ff75c7ed2d9b48aadc"
main_speaker: "Elisabeth Pierre"
title: "Et si la bière était féminine ? | Elisabeth Pierre | TEDxToulouse"
details: "Cette présentation a été faite lors d'un événement TEDx local, produit..."
> tags: Array
```

Campi mancanti dopo la descrizione

04

PySpark Job

Watch next e Count watch next

Il primo job PySpark che abbiamo realizzato serve a creare le collezioni per identificare ogni video.

- Il campo *watch_next_s* indica i video consigliati come successivi al video in questione.
- Il campo *count_wn* contiene il numero di volte in cui il video è consigliato negli altri come successivo.

```
##### CREATE THE AGGREGATE MODEL, ADD WATCH_NEXT TO TEDX_DATASET
watch_next_dataset_agg_a  = watch_next_dataset.groupBy(col("watch_next_idx")).agg({'watch_next_idx':'count'}).withColumnRenamed("count(watch_next_idx)", "count_wn")
watch_next_dataset_agg_b = watch_next_dataset.groupBy(col("idx").alias("idx_ref")).agg(collect_list("watch_next_idx").alias("watch_next_s"))
watch_next_dataset_agg_2 = watch_next_dataset_agg_a.join(watch_next_dataset_agg_b, watch_next_dataset_agg_a.watch_next_idx == watch_next_dataset_agg_b.idx_ref, "left")\
    .drop("watch_next_idx") \
```

05

Collection: *tedx_data_total*

- Ogni documento della collezione identifica un video attraverso il suo ID

- Il campo *count_wn* contiene il numero di volte che un video viene richiamato da un altro come video successivo

06

PySpark Job

Categories

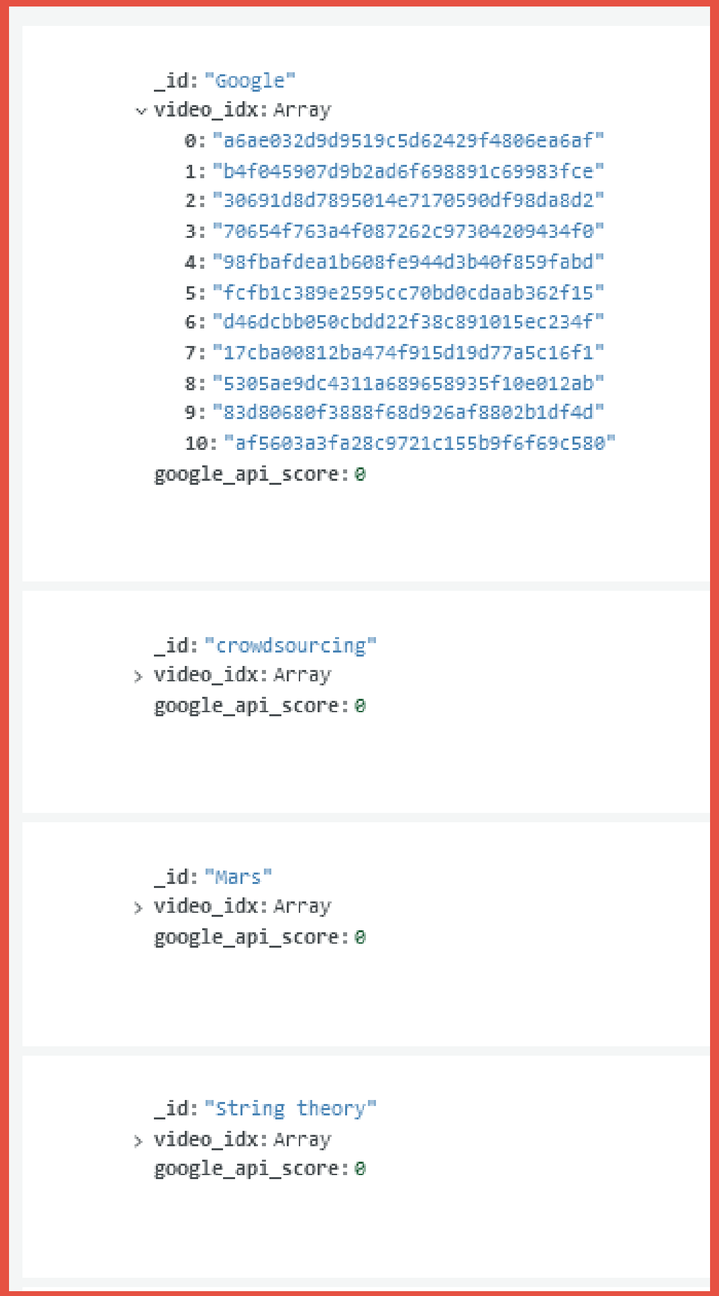
Il secondo job che abbiamo realizzato ha lo scopo di creare una collezione in cui ad ogni categoria/tag vengono associati:

- I video corrispondenti
- Il punteggio di tendenza (per ora non implementato, funzione Batch giornaliera)

```
##### CREATE THE AGGREGATE MODEL, GROUP CATEGORIES
tags_dataset_agg = tags_dataset.groupBy(col("tag").alias("_id")).agg(collect_list("idx").alias("video_idx"))
tags_dataset_agg.printSchema()

##### ADD GOOGLE API FIELD
tags_dataset_agg = tags_dataset_agg.withColumn("google_api_score", lit(0))
```

07



Collection: *tedx_data_categories*

- Ogni documento della collezione identifica una Categoria/Tag attraverso il suo ID
- L'array *video_idx* contiene tutti i video appartenenti alla categoria/tag
- Il campo *google_api_score* rileva lo score di ogni categoria

↓

Campo che aggiorneremo con una funzione batch che giornalmente chiede a Google Trends API la popolarità della categoria e le assegna uno score da 0 a 100

Criticità tecniche

- Una possibile criticità tecnica risiede nello scraper usato per ottenere i dataset. Come evidenziato in precedenza, se nella descrizione di un video sono presenti caratteri speciali come l'escape '\n', vengono create nuove righe nel file .csv che causano dei dati non validi salvati nelle collezioni.

- Inoltre, per decidere l'appartenenza di un video ad una certa categoria utilizziamo i suoi tag, ma questo metodo introduce una certa imprecisione.

09

Possibili evoluzioni

- Unire le categorie simili, magari che variano di qualche carattere ma si riferiscono allo **stesso contesto** come ad esempio:

  TEDx, TED, TEDxTalk...
  Google, Google Search...
  Social, Social media...

  Inoltre unire tutte le categorie stesse categorie che vengono divisi a causa del **case sensitive**.

- Ordinamento documenti nella collezione Tedx_data_categories a seconda del campo google API score

Links

GitHub Repository: https://github.com/samuelexferri/unibg-cloudmobile

Trello Board: https://trello.com/b/iEHLw5as/tcm-1