UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

MyTEDx

Filippo Barbieri - 1054060
Francesco Satriani - 1051361
Andrea Uberti - 1052992

Scopo

Il nostro scopo è stato quello di ottenere una nuova collezione di dati contenente un ulteriore array nel quale viene specificata la lista dei talks correlati al talk appena visto.

Procedimento

Per prima cosa abbiamo caricato sul bucket di S3 il file *watch_next_dataset.csv* contenente i talks correlati per ogni video sulla base dell'id.

Successivamente abbiamo creato un nuovo job che impostasse la nuova collezione di dati. Nello specifico vogliamo avere per ogni talk le seguenti informazioni :

- Id
- Presentatore
- Titolo
- Dettagli
- Data di pubblicazione
- Url
- Tags
- url dei video correlati

Job PySpark

I dati che abbiamo utilizzato sono: la lista dei talks, i tag e la lista dei watch next. Tramite una query ben strutturata siamo riusciti a creare una collezione di dati contenente i video descritti dai tag alla quale appartengono e i relativi video correlati (i watch next).

```
# Carico il file csv che contiene i watch next
wn_dataset_path = "s3://mytedx-data/watch_next_dataset.csv"
# Leggo il file
wn_dataset = spark.read.option("header","true").csv(wn_dataset_path)

# Selezioni dall'intero dataset solo la colonna dell'id del video attuale e l'URL del watch next
wn_dataset_agg=wn_dataset.select(col("idx"),col("url")).groupBy(col("idx").alias("idx_wn")).agg(collect_list("url").alias("watch_next"))

wn_dataset_agg.printSchema()

# Il dataset risultato sarà la join tra le tabelle dei tag, del dataset di tutti i video e dei watch next
tedx_dataset_agg = tedx_dataset.join(tags_dataset_agg, tedx_dataset.idx == tags_dataset_agg.idx_ref, "left") \
    .join(wn_dataset_agg, tedx_dataset.idx == wn_dataset_agg.idx_wn, "left") \
    .drop("idx_ref") \
    .drop("idx_wn") \
    .select(col("idx").alias("_id"), col("*")) \
    .drop("idx")
```

Criticità tecniche

Qualora verranno caricati nuovi video sull'applicazione MyTedx si deve gestire l'aggiornamento automatico relativo alla lista di video correlati per ogni talk.
In questa implementazione non vengono gestiti i duplicati dei video correlati cosa che ha portato ad avere un database con una dimensione in termini di spazio di archiviazione relativamente elevato.

Possibili evoluzioni

Nelle prossime implementazioni abbiamo programmato di gestire l’aggiornamento automatico dei video oltre alla eliminazione dei talks duplicati in modo da risolvere il problema della dimensione del database.
In aggiunta gestiremo anche la cronologia dei video visualizzati.