TEDBooks4You

books worth spreading

Powered by:

TEDx unibg

1081058 - Bellosi Jacopo
1078817 - Poloni Luca

PySpark Job - 1

Il primo Job scritto che abbiamo realizzato va a comporre una collezione con tutti i video del dataset e associa ad ognuno i video correlati. Questo è reso possibile dall'id in comune nel documento "related_videos". Tramite AWS Glue vengono scritti su Mongodb i risultati ottenuti.

```
#AGGIUNGIAMO ORA I VIDEO CORRELATI PER OGNUNO

related_videos_path = "s3://tedx-bellosi-2024-data/related_videos.csv"
related_dataset = spark.read \
    .option("header","true") \
    .option("quote", "\"") \
    .option("escape", "\"") \
    .csv(related_videos_path)

related_dataset = related_dataset.groupBy(col("id").alias("id_from")).agg(collect_list("related_id") \
    .alias("id_related_list"),collect_list("title").alias("title_related_list"))
#EFFETTUIAMO LA JOIN
tedx_dataset_main = tedx_dataset_main.join(related_dataset, tedx_dataset_main.id == related_dataset.id_from, "left") \
    .drop("id_from") \
    .select(col("id").alias("id_related"), col("*"))
#stampiamo a video
related_dataset.printSchema()

tedx_dataset_main.printSchema()

## READ TAGS DATASET
tags_dataset_path = "s3://tedx-bellosi-2024-data/tags.csv"
tags_dataset = spark.read.option("header","true").csv(tags_dataset_path)


# CREATE THE AGGREGATE MODEL, ADD TAGS TO TEDX_DATASET ( Prendo tutti i tag specificati)
tags_dataset_agg = tags_dataset.groupBy(col("id").alias("id_ref")).agg(collect_list("tag").alias("tags"))
tags_dataset_agg.printSchema()
tedx_dataset_agg = tedx_dataset_main.join(tags_dataset_agg, tedx_dataset.id == tags_dataset_agg.id_ref, "left") \
    .drop("id_ref") \
    .select(col("id").alias("_id"), col("*")) \
    .drop("id") \

tedx_dataset_agg.printSchema()


write_mongo_options = {
    "connectionName": "Progetto2",
    "database": "unibg_tedx_2024",
    "collection": "tedx_data",
    "ssl": "true",
    "ssl.domain_match": "false"}
from awsglue.dynamicframe import DynamicFrame
tedx_dataset_dynamic_frame = DynamicFrame.fromDF(tedx_dataset_agg, glueContext, "nested")

glueContext.write_dynamic_frame.from_options(tedx_dataset_dynamic_frame, connection_type="mongodb", connection_options=write_mongo_options)
```

DATA CLEANING

Abbiamo pulito il database eliminando tutti i video che presentano un id nullo.
Abbiamo fatto un left join fra il main dataset e il nuovo dataset creato con gli id nulli ottenendo un dataset con le stesse colonne del main ma ripulito dagli id nulli.

```
count_items = tedx_dataset.count()
count_items_null = tedx_dataset.filter("id is not null").count()


details_dataset_path = "s3://tedx-bellosi-2024-data/details.csv"
details_dataset = spark.read \
    .option("header","true") \
    .option("quote", "\"") \
    .option("escape", "\"") \
    .csv(details_dataset_path)

details_dataset = details_dataset.select(col("id").alias("id_ref"),
                                         col("description"),
                                         col("duration"),
                                         col("publishedAt"))

# FACCIAMO UN JOIN TRA LE DUE TABELLE PER OTTENERE TUTTI I VIDEO CON I RELATIVI DATI
tedx_dataset_main = tedx_dataset.join(details_dataset, tedx_dataset.id == details_dataset.id_ref, "left") \
    .drop("id_ref")
```

Esempio di visualizzazione di documento nella collection tedx_data, viene aggiunto grazie al Job PySpark la lista degli ID correlati...


```
_id: "522612"
id_related : "522612"
slug : "madison_mohns_ai_and_the_paradox_of_self_replacing_workers"
speakers : "Madison Mohns"
title : "AI and the paradox of self-replacing workers"
url : "https://talkstar-photos.s3.amazonaws.com/uploads/d5f40fdf-1911-49ce-95…"
description : "As companies introduce AI into the workplace to increase productivity,…"
duration : "557"
publishedAt : "2024-03-20T14:44:48Z"
▾ id_related_list : Array (6)
    0: "124190"
    1: "89992"
    2: "121579"
    3: "2841"
    4: "17851"
    5: "20368"
```


mongoDB. Atlas

TEDxData

… e anche le liste dei titoli e tags dei video consigliati


```
▾ title_related_list : Array (6)
    0: "Leadership in the age of AI"
    1: "Why people and AI make good business partners"
    2: "What will happen to marketing in the age of AI?"
    3: "How AI can enhance our memory, work and social lives"
    4: "How to get empowered, not overpowered, by AI"
    5: "How AI can save our humanity"
▾ tags : Array (5)
    0: "computers"
    1: "work"
    2: "AI"
    3: "machine learning"
    4: "ethics"
```

Implementazione job dedicato a TEDBooks4you

01

Analizzato le funzionalità e richieste necessarie da utilizzare poi in fututo

02

Creato una collezione con tutti i tag utilizzati e conteggiati

03

Ampliato la collezione con i video da cui sono stato estrapolati i tag, così da fare un’analisi più precisa

PySpark Job - 2

Nel secondo job abbiamo eseguito un conteggio dei tag appartenenti al dataset di tutti i tag dei video. Li abbiamo ordinati in ordine decrescente e successivamente scritti su MongoDB attraverso AWS Glue.

```
# Leggi il dataset dei tag
tags_dataset_path = "s3://tedx-bellosi-2024-data/tags.csv"
tags_dataset = spark.read.option("header", "true").csv(tags_dataset_path)

# Calcola la frequenza dei tag
tag_counts = tags_dataset.groupBy("tag").agg(count("*").alias("tag_count"))

# Ordina i tag per frequenza decrescente
tag_counts = tag_counts.orderBy(col("tag_count").desc())

# Mostra i risultati
tag_counts.show()


# Esegui una join tra i tag più frequenti e il dataset originale dei tag per ottenere tutte le informazioni
top_tags_info = tag_counts.join(tags_dataset, "tag", "left")

# Mostra i risultati con tutte le informazioni
top_tags_info.show()
#---------------
# Converte il DataFrame Spark in un DynamicFrame
tag_counts_dynamic_frame = DynamicFrame.fromDF(tag_counts, glueContext, "tag_counts_dynamic_frame")
```

PySpark Job - 2.1

In questa versione del job abbiamo eseguito una join tra i dataset dei tag e dei video per ottenere gli ID e i titoli dei video associati a ciascun tag. Abbiamo raggruppato i dati per tag e creato una struttura dati all'interno di ciascun tag che include il conteggio dei video associati e una lista di coppie di ID e titoli dei video. Mostriamo in ordine decrescente i risultati e li scriviamo direttamente su MongoDB tramite AWS Glue.

```
# Leggi il dataset dei tag
tags_dataset_path = "s3://tedx-bellosi-2024-data/tags.csv"
tags_dataset = spark.read.option("header", "true").csv(tags_dataset_path)

# Leggi il dataset principale dei video
tedx_dataset_path = "s3://tedx-bellosi-2024-data/final_list.csv"
tedx_dataset = spark.read.option("header", "true").csv(tedx_dataset_path)

# Esegui una join tra i tag e i video per ottenere gli ID e i titoli dei video per ciascun tag
tag_video_info = tags_dataset.join(tedx_dataset, tags_dataset.id == tedx_dataset.id, "left") \
    .select(tags_dataset["tag"], tedx_dataset["id"].alias("video_id"), tedx_dataset["title"].alias("video_title"))

# Raggruppa per tag e crea una struttura dati all'interno di ciascun tag
tag_info = tag_video_info.groupBy("tag") \
    .agg(count("*").alias("tag_count"), collect_list(struct(col("video_id"), col("video_title"))).alias("videos"))

# Mostra i risultati
tag_info = tag_info.orderBy(col("tag_count").desc())

#FINE PARTE AGGIUNTA PER ORDINARE
tag_info.show(truncate=False)
```

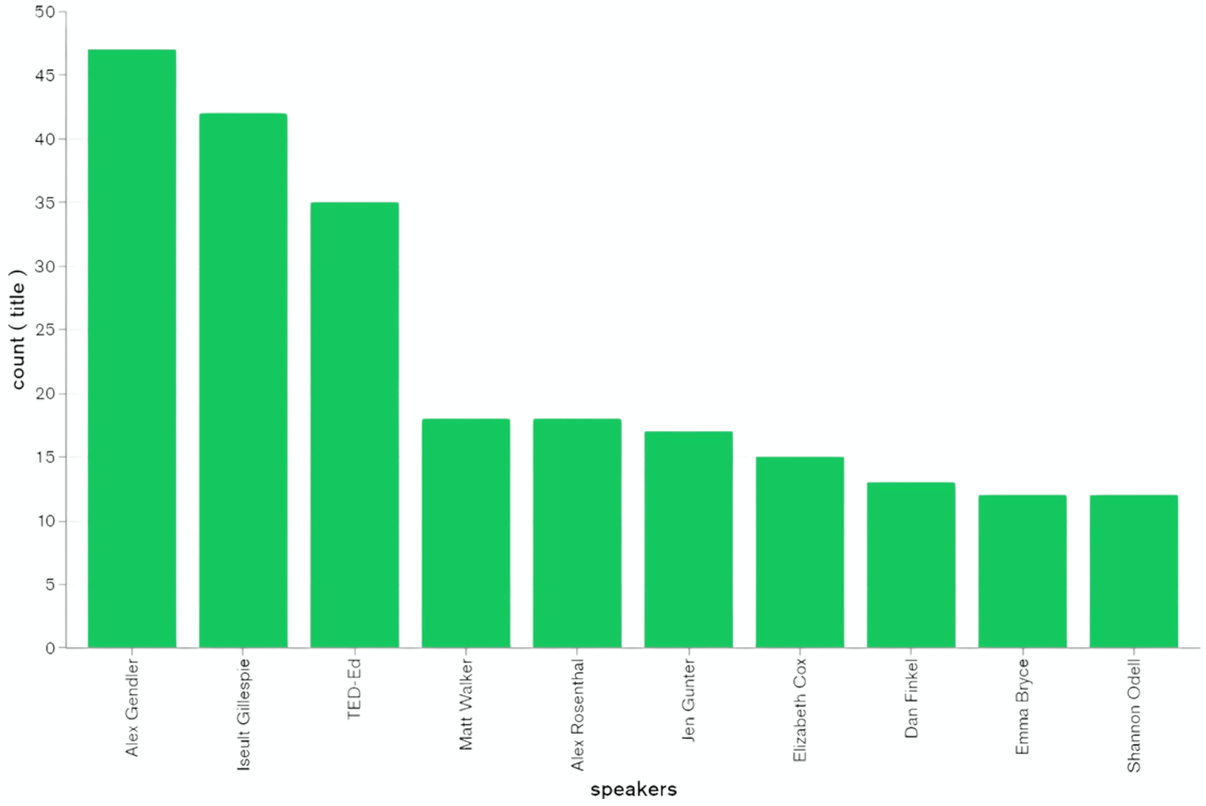
Results Job - 2

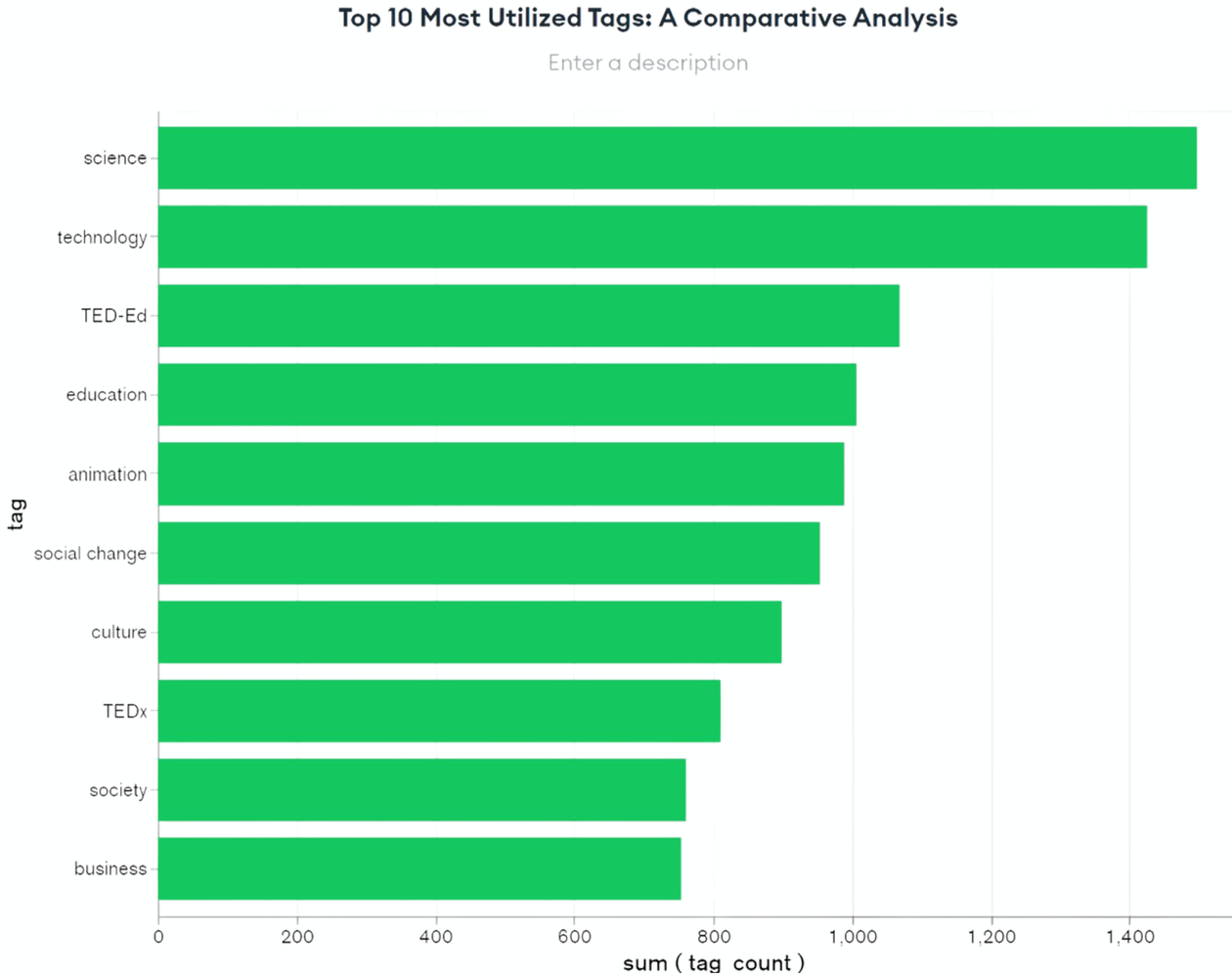

```
_id: ObjectId('663a53d18fd35c2abb1fa931')
tag : "leadership"
tag_count : 254

_id: ObjectId('663a53d18fd35c2abb1fa916')
tag : "drugs"
tag_count : 49

_id: ObjectId('663a53d18fd35c2abb1fa919')
```


Results Job - 2.1


```
_id: ObjectId('663a5a48b4e3eb14c9498641')
tag : "wind energy"
tag_count : 16
▾ videos : Array (16)
  ▸ 0: Object
  ▸ 1: Object
  ▸ 2: Object
  ▾ 3: Object
      video_id : "419544"
      video_title : "How wind energy could power Earth ... 18 times over"
  ▸ 4: Object
  ▸ 5: Object
  ▸ 6: Object
  ▸ 7: Object
```

Visualizzazione grafica dei risultati





Criticità e possibili evoluzioni

Generalizzazione dei tag, impossibilità di essere verticali nel suggerimento dei libri con questi dati

Web Scraping per offrire dei tag più personalizzati in base alle parole frequenti utilizzate nei video

TED Books4You

BOARD

GIT