

# TEDx TOK

Progetto parte 2:  
PySpark

Mazzoleni Gabriele – 1079514  
Masinari Gabriele - 1079692

# Job PySpark (MyTedx): Load\_data

Realizzato tramite AWS Glue con un job PySpark, abbiamo aggregato i dati dal file "related\_videos" per poi inserirli nella collezione "tedx\_data".

Abbiamo aggiunto a ogni documento un array di ID dei video correlati, dei loro titoli e dei loro relatori.

Abbiamo realizzato anche un altro job, chiamato "TedTok\_data\_loader", che esegue una simile operazione, creando una collezione con i dati utili ai fini dell'applicazione TedTok (gli stessi, senza le immagini).

Questo nuovo dato ci permetterà, all'interno della nostra applicazione, di consigliare all'utente video simili a quelli da lui visionati, aumentando così l'utilizzo del nostro software.

# Codice aggiunto al job:

## Load\_data

```
# ADDITION OF "WATCH NEXT" LINKS TO DATASET
related_dataset_path = "s3://tedx-2024-data-mazzoleni-g/related_videos.csv"
related_dataset = details_dataset = spark.read.option("header","true").csv(related_dataset_path)

related_dataset_with_urls=related_dataset.join(tedx_dataset, related_dataset.internalId == tedx_dataset.id, "left") \
    .select(related_dataset["id"], related_dataset["related_id"], related_dataset["title"], related_dataset["presenterDisplayName"], tedx_dataset["url"].alias("related_url"))

#AGGREGATE WATCH NEXT DATA AND JOIN TO MAIN DATASET
related_dataset_agg = related_dataset_with_urls.groupBy(col("id").alias("id_ref")) \
    .agg(collect_list(struct(col("related_id").alias("related_video_ids"), col("title").alias("related_video_title"),
    col("presenterDisplayName").alias("related_presentedBy"), col("related_url"))).alias("Related_videos"))

tedx_dataset_with_watch_next = tedx_dataset_full.join(related_dataset_agg, tedx_dataset_full.id == related_dataset_agg.id_ref, "left") \
    .drop("id_ref") \
    .select(col("_id"), col("*")) \

tedx_dataset_with_watch_next.printSchema()

write_mongo_options = {
    "connectionName": "TEDx 2024 by GabTheBest",
    "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_with_watch_next, glueContext, "nested")

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

# Esempio di dato in MongoDB

```
{
  "_id": "526880",
  "slug": "george_zaidan_how_do_gas_masks_actually_work",
  "speakers": "George Zaidan",
  "title": "How do gas masks actually work?",
  "url": "https://www.ted.com/talks/george_zaidan_how_do_gas_masks_actually_work",
  "description": "You might think of gas masks as clunky military-looking devices. But i_",
  "duration": "254",
  "publishedAt": "2024-04-30T15:14:51Z",
  "tags": Array (8),
  "imageUrl": "https://talkstar-assets.s3.amazonaws.com/production/talks/talk_128547/_",
  "Related_videos": Array (3)
    0: Object
      related_video_ids: "109914"
      related_video_title: "Whatever happened to the hole in the ozone layer?"
      related_presentedBy: "Stephanie Honchell Smith"
    1: Object
    2: Object
}
```

# Job PySpark (TedTok): TedTok\_Tag\_data

Realizzata tramite AWS Glue con un job PySpark.

Questo job compila l'elenco dei tag disponibili associando ciascuno la lista dei talk che lo includono;

Lo scopo di questo job è mostrare inizialmente all'utente la lista di tutti i tag disponibili nel dataset. Successivamente, una volta selezionati i tag desiderati, l'utente potrà visionare i video corrispondenti ai tag scelti.

# Codice del Job PySpark: TedTok\_Tag\_data

```
import sys
import json
import pyspark
from pyspark.sql.functions import col, collect_list, array_join, struct, array_distinct

from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)
job.commit()

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

# Codice del Job PySpark: TedTok\_Tag\_data

```
# LOAD TALK DATA
tedx_dataset_path = "s3://tedx-2024-data-mazzoleni-g/final_list.csv"
tedx_dataset = spark.read \
    .option("header", "true") \
    .option("quote", "\"") \
    .option("escape", "\\") \
    .csv(tedx_dataset_path)
tedx_dataset.printSchema()

tags_dataset_data = tags_dataset.join(tedx_dataset, tags_dataset.id == tedx_dataset.id, "left") \
    .select(tags_dataset["tag"], tedx_dataset["id"].alias("talk_id"), tedx_dataset["title"].alias("talk_title"), tedx_dataset["url"].alias("talk_url"))

# CREATE THE AGGREGATE MODEL, ADD TAGS TO TEDX_DATASET
tags_dataset_agg = tags_dataset_data.groupBy(col("tag").alias("id")) \
    .agg(array_distinct(collect_list(struct(col("talk_id"), col("talk_title"), col("talk_url")))).alias("related_talks"))
tags_dataset_agg.printSchema()

# CREATE A NEW MONGODB COLLECTION
write_mongo_options = {
    "connectionName": "TEDx 2024 by GabTheBest",
    "database": "unibg_tedx_2024",
    "collection": "TedTok_tags",
    "ssl": "true",
    "ssl.domain_match": "false"}
from aws glue.dynamicframe import DynamicFrame
tags_dataset_dynamic_frame = DynamicFrame.fromDF(tags_dataset_agg, glueContext, "nested")

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

array\_distinct  
è stato  
utilizzato  
per togliere  
gli eventuali  
video  
duplicati

# Esempio di dato in MongoDB

```
_id: "TED Membership"  
related_talks: Array (38)
```

---

```
_id: "evolution"  
related_talks: Array (176)
```

---

```
_id: "wildlife"  
related_talks: Array (1)  
  0: Object  
    talk_id: "524879"  
    talk_title: "Let your garden grow wild"  
    talk_url: "https://www.ted.com/talks/rebecca_mcmackin_let_your_garden_grow_wild"
```

---



# Criticità



Alto livello di duplicazione dei dati dei talk(nella collezione dei tags).

Inserimento di nuovi talk potrebbe richiedere la ricostruzione di tutte le collezioni

# Vantaggi



Facilitazione degli interessi dell'utente nella ricerca di talk che lo interessano.

Facilitazione all'accesso dei talk correlati, sia per ID che per tag.

# TEDx TOK

[Trello board](#)

[GitHub](#)