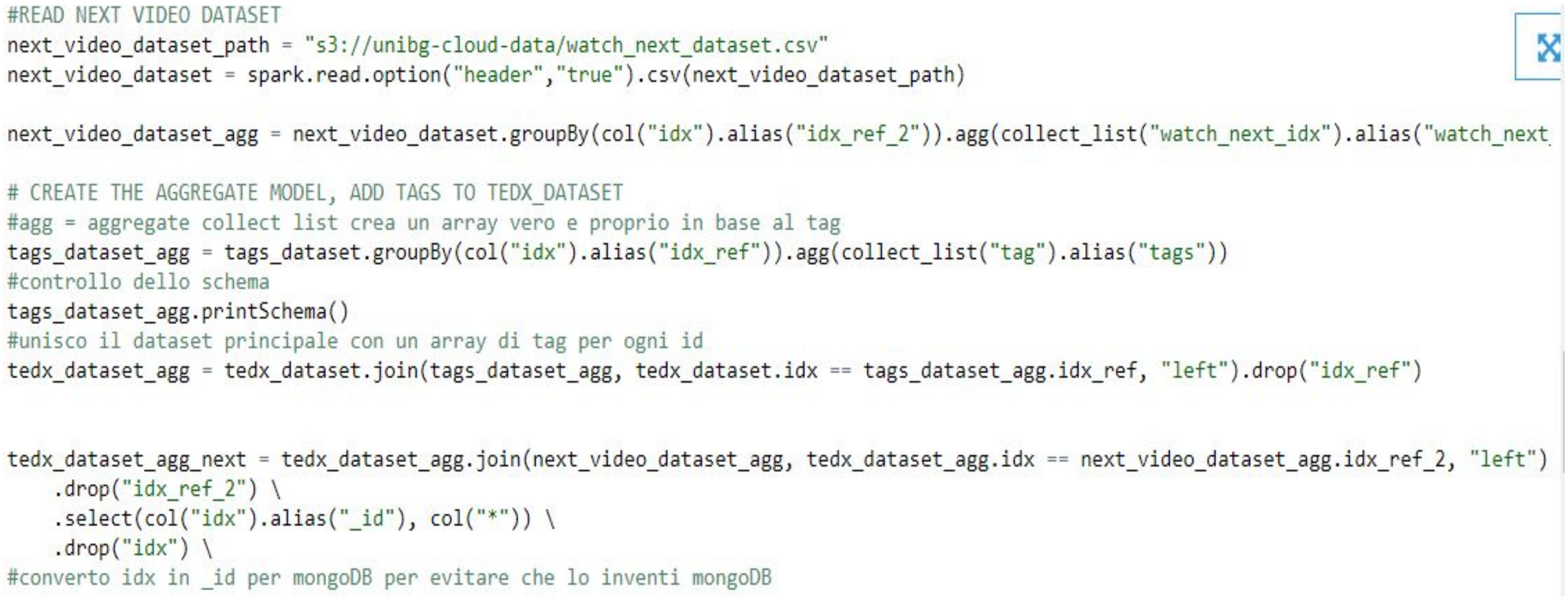
TEDdy

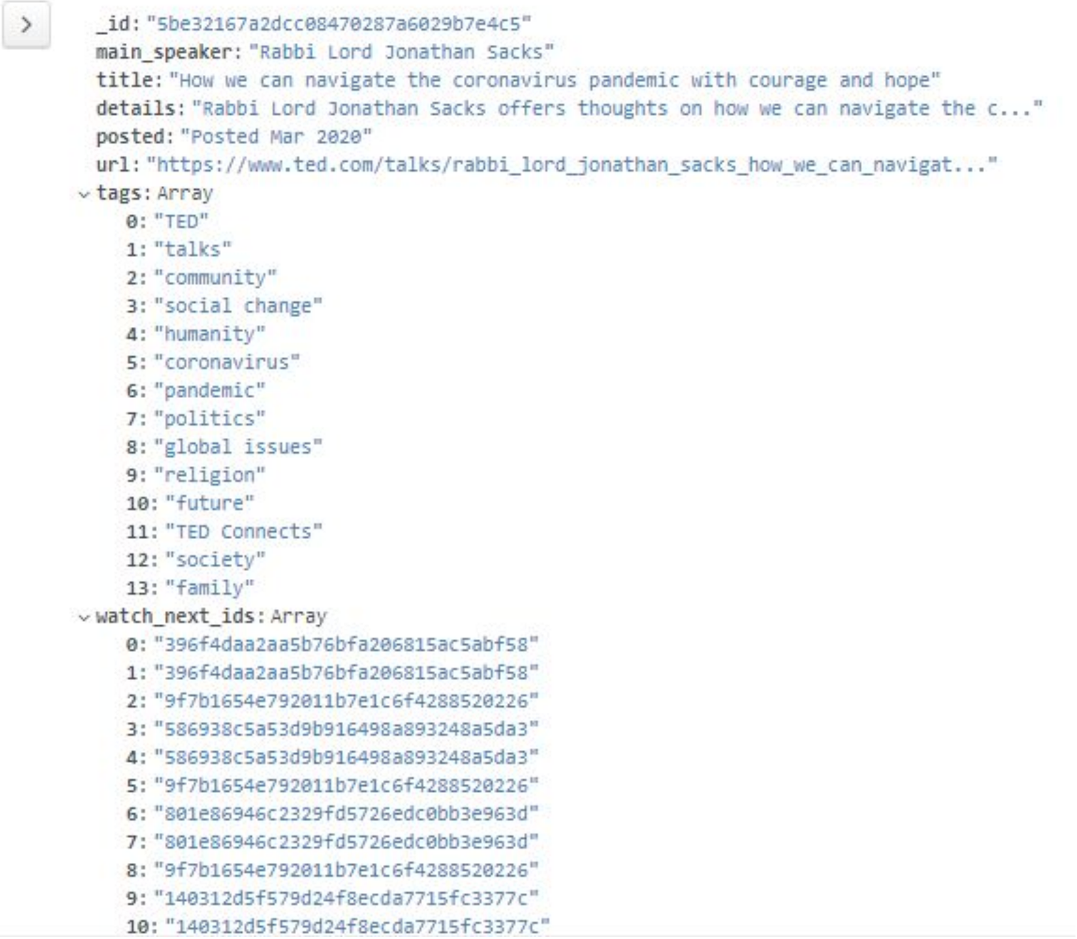
Calegari Andrea 1041183
Polver Marco 1040386
Del Prete Giovanni 1035205

Watch next code



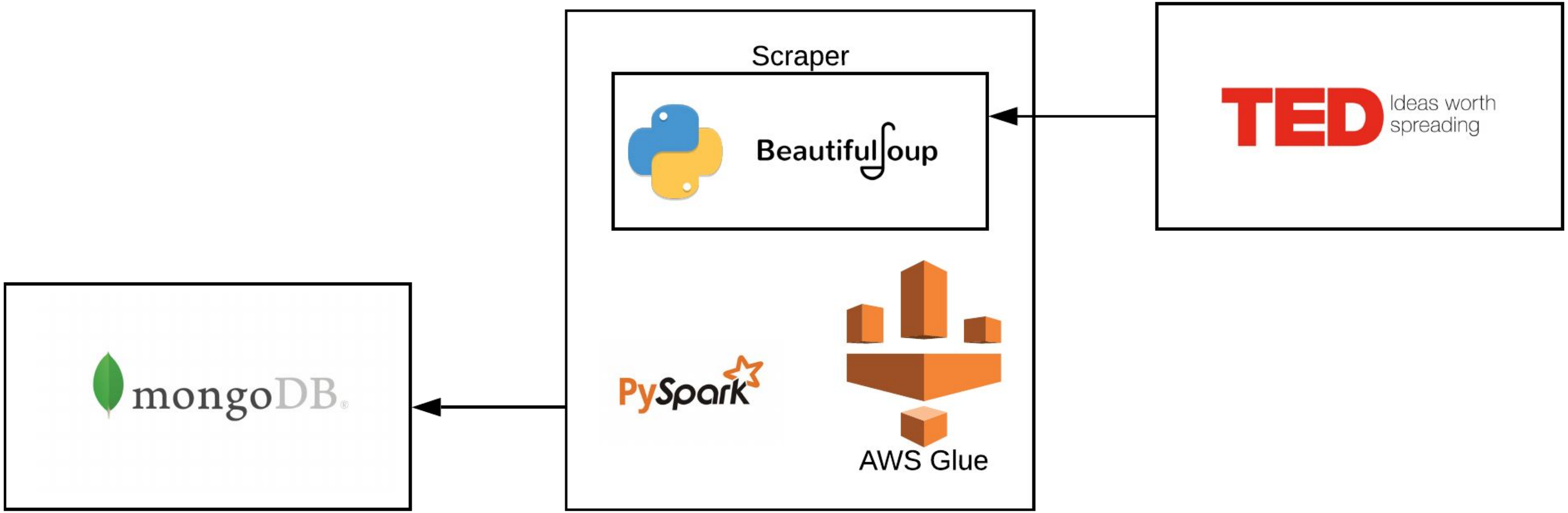
```
#READ NEXT VIDEO DATASET
next_video_dataset_path = "s3://unibg-cloud-data/watch_next_dataset.csv"
next_video_dataset = spark.read.option("header","true").csv(next_video_dataset_path)

next_video_dataset_agg = next_video_dataset.groupBy(col("idx").alias("idx_ref_2")).agg(collect_list("watch_next_idx").alias("watch_next

# CREATE THE AGGREGATE MODEL, ADD TAGS TO TEDX_DATASET
#agg = aggregate collect list crea un array vero e proprio in base al tag
tags_dataset_agg = tags_dataset.groupBy(col("idx").alias("idx_ref")).agg(collect_list("tag").alias("tags"))
#controllo dello schema
tags_dataset_agg.printSchema()
#unisco il dataset principale con un array di tag per ogni id
tedx_dataset_agg = tedx_dataset.join(tags_dataset_agg, tedx_dataset.idx == tags_dataset_agg.idx_ref, "left").drop("idx_ref")


tedx_dataset_agg_next = tedx_dataset_agg.join(next_video_dataset_agg, tedx_dataset_agg.idx == next_video_dataset_agg.idx_ref_2, "left")
    .drop("idx_ref_2") \
    .select(col("idx").alias("_id"), col("*")) \
    .drop("idx") \
#converto idx in _id per mongoDB per evitare che lo inventi mongoDB
```

Watch next result



```
_id: "5be32167a2dcc08470287a6029b7e4c5"
main_speaker: "Rabbi Lord Jonathan Sacks"
title: "How we can navigate the coronavirus pandemic with courage and hope"
details: "Rabbi Lord Jonathan Sacks offers thoughts on how we can navigate the c..."
posted: "Posted Mar 2020"
url: "https://www.ted.com/talks/rabbi_lord_jonathan_sacks_how_we_can_navigat..."
tags: Array
    0: "TED"
    1: "talks"
    2: "community"
    3: "social change"
    4: "humanity"
    5: "coronavirus"
    6: "pandemic"
    7: "politics"
    8: "global issues"
    9: "religion"
    10: "future"
    11: "TED Connects"
    12: "society"
    13: "family"
watch_next_ids: Array
    0: "396f4daa2aa5b76bfa206815ac5abf58"
    1: "396f4daa2aa5b76bfa206815ac5abf58"
    2: "9f7b1654e792011b7e1c6f4288520226"
    3: "586938c5a53d9b916498a893248a5da3"
    4: "586938c5a53d9b916498a893248a5da3"
    5: "9f7b1654e792011b7e1c6f4288520226"
    6: "801e86946c2329fd5726edc0bb3e963d"
    7: "801e86946c2329fd5726edc0bb3e963d"
    8: "9f7b1654e792011b7e1c6f4288520226"
    9: "140312d5f579d24f8ecda7715fc3377c"
    10: "140312d5f579d24f8ecda7715fc3377c"
```

Scraper


Scraper
Beautiful Soup
TED Ideas worth spreading
PySpark
AWS Glue
mongoDB

Scraper

- Utilizzo di BeautifulSoup invece di Selenium, vista l’impossibilità di avviare un browser in AWS Glue
- Problema: impossibilità da parte di BeautifulSoup di accedere a codice HTML generato da ReactJS → impossibilità di accedere ai dati riguardanti i “next watch”

PROBLEM SOLVED

Utilizziamo tecniche di ML per consigliare il prossimo video

Scraper

- Problema: difficoltà nella creazione di file all'interno di un bucket S3 da job AWS Glue
- Appesantimento del sistema (scrittura su file csv e successiva lettura del file per aggiornamento DB)

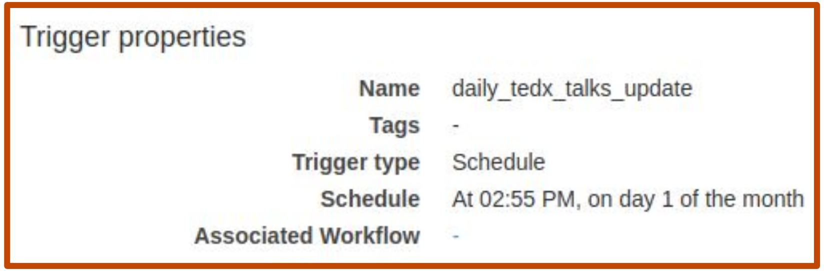
↓

PROBLEM SOLVED

Caricamento diretto dei risultati su Atlas MongoDB

Scraper

- Trigger mensile per l’aggiornamento del DB. La frequenza mensile è stata una scelta cost-driven, ma in produzione potremmo impostare una frequenza maggiore



Trigger properties

Name	daily_tedx_talks_update
Tags	-
Trigger type	Schedule
Schedule	At 02:55 PM, on day 1 of the month
Associated Workflow	-

Scraper

Link allo script: https://github.com/cale96/tecnologie_cloud_mobile/blob/master/scraper/scraper.py