GeoTEDx

Matteo Cesari (MAT. 1073570)

Davide Girolamo (MAT. 1073645)

GitHub: https://github.com/TurnTheBait/GeoTEDx

Job AWS Glue

Job Create_Data_Lake

Lo scopo principale dello script è elaborare, aggregare e archiviare i dati dei dataset TEDx su MongoDB, consentendo un'analisi e una consultazione più efficienti.

- Data cleaning: rimozione dei dati nulli e duplicati per tutti i dataset.
- PySpark Job: creazione delle collections per identificare ogni video.

```
##### READ TAGS DATASET
tags_dataset_path = "s3://bucket-dati-test1-2023/tags_dataset.csv"
tags_dataset = spark.read.option("header","true").csv(tags_dataset_path)

##### CLEAN TAGS DATASET
print(f"TOTAL TAGS DATASET: {tags_dataset.count()}")
tags_dataset = tags_dataset.dropDuplicates()
#### REMOVE DUPLICATES
print(f"TAGS DATASET without DUPLICATES {tags_dataset.count()}")
```

```
##### CLEAN TEDX DATASET
print(f"TOTAL DATASET IDX: {tedx_dataset.count()}")
tedx_dataset = tedx_dataset.filter(length("idx") == 32)     #### REMOVE INVALID IDX
print(f"DATASET without INVALID IDX: {tedx_dataset.count()}")

tedx_dataset = tedx_dataset.dropDuplicates()  #### REMOVE DUPLICATES
print(f"DATASET without DUPLICATES: {tedx_dataset.count()}")

##### FILTER ITEMS WITH NULL POSTING KEY
count_items = tedx_dataset.count()
count_items_null = tedx_dataset.filter("idx is not null").count()

print(f"Number of items from RAW DATA {count_items}")
print(f"Number of items from RAW DATA with NOT NULL KEY {count_items_null}")
```

Job AWS Glue

Job Watch_Next

Lo scopo principale dello script è quello di incorporare all'interno della documentazione su MongoDB di ogni talk, l'array watch_next contenente gli id dei video consigliati.

- Data cleaning: rimozione dei dati nulli e duplicati per tutti i dataset.

```
##### READ WATCH NEXT DATASET
watch_next_dataset_path = "s3://bucket-dati-test1-2023/watch_next_dataset.csv"
watch_next_dataset = spark.read.option("header","true").csv(watch_next_dataset_path)

##### CLEAN WATCH NEXT DATASET
print(f"TOTAL WATCH NEXT DATASET: {watch_next_dataset.count()}")
watch_next_dataset = watch_next_dataset.dropDuplicates()    #### REMOVE DUPLICATES
print(f"WATCH NEXT DATASET without DUPLICATES {watch_next_dataset.count()}")
```

- Aggregate model: Creazione del modello aggregato aggiungendo i dati "watch_next" al dataset aggregato TEDx e Tags.

```
# CREATE THE AGGREGATE MODEL, ADD TAGS TO TEDX_DATASET
tags_dataset_agg = tags_dataset.groupBy(col("idx").alias("idx_ref")).agg(collect_list("tag").alias("tags"))
tags_dataset_agg.printSchema()
tedx_dataset_agg = tedx_dataset.join(tags_dataset_agg, tedx_dataset.idx == tags_dataset_agg.idx_ref, "left") \
    .drop("idx_ref") \
    .select(col("idx").alias("_id"), col("*")) \
    .drop("idx") \

tedx_dataset_agg.printSchema()
```

Collection su MongoDB

Collection: **tedx_data**

- Ogni documento della collezione rappresenta un talk. Ogni elemento è identificato da in id univoco, vengono inoltre forniti varie informazioni riguardo il contenuto e le visualizzazioni del video.
- I **tag** associati al talk vengono inseriti in un array.
- L'array **watch_next** contiene gli id dei talk indicati come "watch_next" del talk considerato.

Sviluppi futuri

- Inserire una maggiore quantità di tag all’interno dei dati di ciascun talk in modo più attinente per migliorare la qualità dei video suggeriti
- Prendere in considerazione la data di pubblicazione dei talk per consigliare video più recenti