



TED Watch Next

Thumchan Tanakon (1059590)
Mazzoleni Paolo (1057949)



Descrizione

Il codice che abbiamo realizzato ha lo scopo di estrarre i dati dal file csv, filtrarli, elaborarli e caricare i risultati su MongoDB.

Oltre al codice di base che ci è stato fornito, abbiamo scritto una nuova parte per aggiungere il campo watch next al dataset. Questo funge da contenitore per vari Object che contengono a loro volta tre campi: id, url e titoli dei video consigliati.

Per raggiungere gli scopi prefissati abbiamo utilizzato:

- AWS S3: il servizio di storage per memorizzare i file csv
- AWS Glue: la piattaforma di tipo serverless per eseguire il job



Criticità

Durante lo sviluppo del codice, le parti in cui abbiamo riscontrato criticità sono:

- File CSV non del tutto corretto - il file contiene degli a capo che causano la lettura errata del dataset. Abbiamo aggiunto option multiline per correggere l'estrazione dei dati dal file CSV.

```
#### READ INPUT FILES TO CREATE AN INPUT DATASET  
tedx_dataset = spark.read \  
    .option("header","true") \  
    .option("quote", "\"") \  
    .option("multiline","true") \  
    .option("escape", "\"") \  
    .csv(tedx_dataset_path)
```



Criticità

Durante lo sviluppo del codice, le parti in cui abbiamo riscontrato criticità sono:

- Link watch next errati - il file contiene dei link non funzionanti. Abbiamo utilizzato un filtro per verificare la correttezza degli URL.

```
watch_next_dataset = watch_next_dataset.filter("url LIKE 'https://www.ted.com/talks/%'" ) # Get only talks url
```

- Dati duplicati - il file contiene dei link duplicati. Abbiamo utilizzato la funzione *dropDuplicates* per cancellarli.

```
watch_next_dataset = watch_next_dataset.dropDuplicates(["idx", "watch_next_idx"]) # Delete duplicated rows
```

- La funzione *groupBy* con una struttura - per aumentare la qualità dei dati abbiamo deciso di strutturare il campo 'watch_next' come una lista di oggetti contenente 3 campi essenziali : id, url e titolo del prossimo video.


```
watch_next_dataset_agg = watch_next_dataset.join(tedx_dataset, tedx_dataset.idx == watch_next_dataset.watch_next_idx, "left") \
    .select(watch_next_dataset["*"],tedx_dataset["title"])

watch_next_dataset_agg = watch_next_dataset_agg.groupBy(col("idx").alias("idx_ref")) \
    .agg(collect_list(struct("watch_next_idx", "url", "title")).alias("watch_next"))
watch_next_dataset_agg.printSchema()
```



Risultato finale

```
_id: "08438d7fc43df207af7e95180f6599e9"
main_speaker: "Péter Fankhauser"
title: "Meet Rezero, the dancing ballbot"
details: "Engineering student Péter Fankhauser demonstrates Rezero, a robot that..."
posted: "Posted Nov 2011"
url: "https://www.ted.com/talks/peter_fankhauser_meet_rezero_the_dancing_bal..."
> tags: Array
✓ watch_next: Array
  ✓ 0: Object
    watch_next_idx: "293befd4e2d163a016e23d91b176aa29"
    url: "https://www.ted.com/talks/rodney_brooks_robots_will_invade_our_lives"
    title: "Robots will invade our lives"
  > 1: Object
  > 2: Object
  > 3: Object
  > 4: Object
  > 5: Object
```



<https://github.com/paolomazzoleni4/TCM2021/blob/main/CreaDataLeakeWithWatchNext.py>