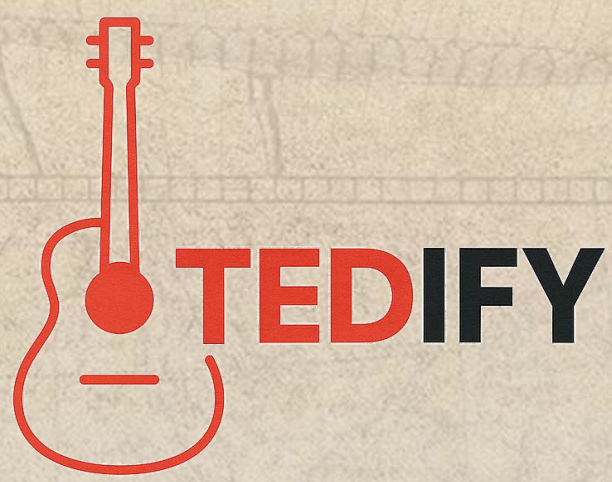
TEDify

«Musica che ispira, idee che suonano»



DAMIANO CARRARA (MATR. 1067871) – HOMEWORK 2

RELEASE PLAN

REPOSITORY

IMPLEMENTAZIONE WATCH_NEXT

- La scelta è stata quella di implementare il watch_next mediante un array contenente gli id dei video correlati.
- La motivazione per questa scelta (rispetto per esempio ad un array di oggetti) è stata il mantenere più snella possibile la struttura del singolo documento. Le informazioni aggiuntive di ogni talk possono essere ottenute cercando il rispettivo id.

JOB ADD_WATCH_NEXT

```
89  #READ RELATED VIDEOS AND REMOVE DUPLICATES
90  related_videos_path = "s3://dc-data-2025/related_videos.csv"
91  related_videos_dataset = spark.read.option("header", "true").csv(related_videos_path).dropDuplicates()
92
93  # Fai un self-join tra related_videos e i video effettivamente presenti nel dataset
94  # cioè: tieni solo i related_id che sono presenti in tedx_dataset_agg._id
95
96  # Prima rinominiamo le colonne per chiarezza
97  related_videos_dataset = related_videos_dataset.withColumnRenamed("id", "main_id").withColumnRenamed("related_id", "related_id_raw")
98
99  # Teniamo solo i related_id che sono presenti nel dataset
100 related_videos_filtered = related_videos_dataset.join(
101     tedx_dataset_agg.select(col("_id").alias("valid_related_id")),
102     related_videos_dataset.related_id_raw == col("valid_related_id"),
103     "inner"
104 ).drop("valid_related_id")
105
106 # Ora possiamo aggregare i related_id filtrati
107 related_videos_dataset_agg = related_videos_filtered.groupBy(col("main_id").alias("id_watch_next")) \
108     .agg(collect_list("related_id_raw").alias("Watch_next_id"))
109
110 #ADD RELATED VIDEOS TO AGGREGATE MODEL
111 tedx_dataset_agg = tedx_dataset_agg.join(
112     related_videos_dataset_agg,
113     tedx_dataset_agg._id == related_videos_dataset_agg.id_watch_next,
114     "left"
115 ).drop("id_watch_next")
```

Codice completo su GitHub

ESEMPIO DI DOCUMENTO SU MONGODB


```
_id: "567142"
slug : "charles_duhigg_the_science_behind_dramatically_better_conversations_ma…"
speakers : "Charles Duhigg"
title : "The science behind dramatically better conversations"
url : "https://www.ted.com/talks/charles_duhigg_the_science_behind_dramatical…"
description : "The key to deeply connecting with others is about more than just talki…"
duration : "707"
publishedAt : "2025-03-06T15:48:43Z"
▾ tags : Array (4)
    0: "science"
    1: "communication"
    2: "personal growth"
    3: "TEDx"
▾ Watch_next_id : Array (1)
    0: "144704"
```

CRITICITÀ TECNICHE

- Le maggiori difficoltà sono legate alla scarsa pulizia dei dati:
  - Innanzitutto abbiamo rimosso i duplicati
  - La maggior parte degli id contenuti nei watch_next rimandavano a video non presenti nel dataset. È stato quindi necessario implementare una parte del codice che mantenesse solo i riferimenti consistenti.
- Debugging complesso: i log di AWS sono risultati un po' confusionari da leggere

TEDIFY_JOB

- Innanzitutto abbiamo filtrato i talk per ottenere solo quelli contenenti sia il tag “music” che il tag “performance”, avendo osservato che questa combinazione di tag corrisponde alle performance musicali dal vivo contenute nel dataset.
- In seguito abbiamo convertito la durata di ogni talk, che nel file .csv era indicata in secondi, in una più classica notazione minuti:secondi, ritenuta più efficace per essere poi visualizzata nell’applicazione.

FILTRAGGIO TAGS

```
117  #FILTRAGGIO TAGS
118  filtered_dataset=tedx_dataset_agg
119  filtered_dataset=tedx_dataset_agg.filter(size(array_intersect(col("tags"),array(lit("music"),lit("performance"))))==2)
120
121  #RIPULISCO ULTERIORMENTE WATCH_NEXT
122  valid_ids_list = [row._id for row in filtered_dataset.select("_id").collect()]
123  valid_ids_col = array([lit(id_val) for id_val in valid_ids_list])
124
125  filtered_dataset = filtered_dataset.withColumn(
126      "Watch_next_id",
127      array_intersect(col("Watch_next_id"), valid_ids_col)
128  )
129
130  print("Schema dopo la pulizia di Watch_next_id:")
131  filtered_dataset.printSchema()
132
133
134  #PRINT FILTERED DATASET
135  filtered_dataset.printSchema()
136  filtered_dataset.show()
137
138  filtered_dataset = filtered_dataset.withColumn("duration_int", col("duration").cast(IntegerType()))
139
```

Codice completo su GitHub

CONVERSIONE DURATA

```python
#CONVERT DURATION INTO MINUTES:SECONDS
def convert_safe(seconds):
    if seconds is None:
        return None
    try:
        seconds = int(seconds)
        minutes = seconds // 60
        seconds %= 60
        return f"{minutes:02d}:{seconds:02d}"
    except Exception as e:
        return None

#convertUDF=udf(lambda m:convert(m))
convertUDF = udf(convert_safe, StringType())
filtered_dataset = filtered_dataset.withColumn("duration", convertUDF(col("duration_int"))).drop("duration_int")
#filtered_dataset_converted=filtered_dataset.select(col("*"),convertUDF(col("duration").alias("duration_Sec")))
filtered_dataset.show()
```


```
_id: "552755"
slug : "xiuhtezcatl_careful_veils"
speakers : "Xiuhtezcatl"
title : ""Careful" / "Veils""
url : "https://www.ted.com/talks/xiuhtezcatl_careful_veils"
description : "Musician Xiuhtezcatl raps, sings and plays piano in a performance seam…"
duration : "10:41"
publishedAt : "2024-12-13T15:37:02Z"
tags : Array (3)
  0: "music"
  1: "performance"
  2: "indigenous peoples"
Watch_next_id : null
```


Esempio di documento nel DB

CRITICITÀ TECNICHE

- Scarsa quantità di talk: una volta filtrati i dati, tenendo solo quelli che presentano i due tags music e performance, si è ottenuto un database di dimensione decisamente ridotta rispetto a quello iniziale.

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
tedify	233	148.27KB	652B	188KB	1	24KB	24KB
tedx_data	7055	5.8MB	862B	6.82MB	1	352KB	352KB

- Inconsistenza dei riferimenti: dopo aver filtrato i talk, si è di nuovo presentato il problema dei riferimenti inconsistenti in watch_next: abbiamo quindi ripulito i dati, con il risultato che però quasi tutti gli array watch_next risultano essere vuoti.
- Ciò è dovuto anche ad una scarsa logica dei riferimenti: per esempio due performance dello stesso artista (Jacob Collier) non sono in relazione tra di loro.

POSSIBILI EVOLUZIONI

- Aggiungere ad ogni documento la relativa immagine di copertina.
- Rendere possibile l'ascolto in modalità podcast/YouTube Music (solo audio, senza video).
- Fornire la possibilità di ascolto in modalità offline, consentendo all'utente di salvare l'audio sul proprio dispositivo.