TRIP WITH TED

TRESOLDI ANDREA [redacted]

1

Create data lake

- Introduzione script e collezioni generate
- Dati utilizzati e pulizia

2

Tedx data collection

- Creazione del modello aggregato
- Modello aggregato finale

3

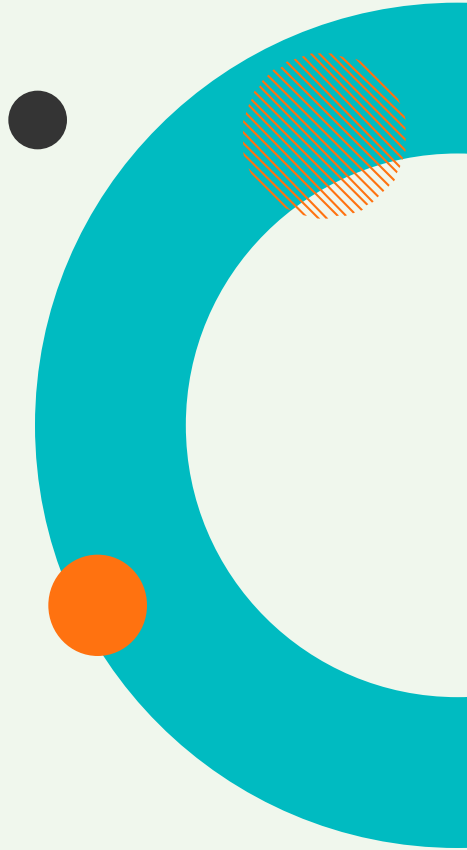
Main speaker data collection

- Creazione del modello aggregato
- Modello aggregato finale

4

Criticità ed evoluzioni

- Possibili criticità ed evoluzioni.



1 – Create data lake

Descrizione:

- L’obiettivo del job pyspark Create Data Lake è quello di creare all’interno del database unibg_tedx_2021 due collezioni:
  - tedx_data_collection.
  - main_speaker_data_collection.

Dati Utilizzati e pulizia:

- Per la realizzazione delle due collezioni sono state utilizzate quattro fonti di dati differenti:
  - tags_dataset.csv
  - tedx_dataset.csv
  -watch_next_dataset.csv
  -main_speakers_dataset.csv

1 – Create data lake

- In particolare il file main_speakers_dataset.csv non è stato fornito dal professore ma è stato generato da un piccolo script Python utilizzando l'API di Wikipedia: il codice recupera, se disponibile, una piccola descrizione di ogni main_speaker presente all'interno del file tedx_dataset.csv.

```
import pandas as pd
import numpy as np
import wikipedia

df = pd.read_csv('tedx_dataset.csv')
main_speakers = df['main_speaker'].tolist()
main_speakers = np.unique(main_speakers)
descriptions = []

print("start")
for main_speaker in main_speakers:
    try:
        title = wikipedia.search(main_speaker)[0]
        summary = wikipedia.summary(title, sentences=1)
    except:
        summary = ''

    descriptions.append({'main_speaker': main_speaker, 'main_speaker_description': summary})

df = pd.DataFrame.from_records(descriptions)
df.to_csv('main_speaker_description_dataset.csv')
print('Done')
```

1 – Create data lake

- I dati utilizzati per creare il DWH avevano alcuni difetti che sono stati ovviamente corretti prima del loro inserimento in banca dati:
- Il file tedx_dataset.csv aveva dei record che occupavano più righe; inoltre sono stati selezionati solo i talks che avessero il campo idx lungo 32 caratteri.
- Invece all'interno del file watch_next_dataset.csv tutti i talks avevano associato watch_next duplicati e un watch_next non valido.

```
tedx_dataset = spark.read \
    .option("header","true") \
    .option("quote", "\"") \
    .option("escape", "\"") \
    .option("multiline", "true") \
    .csv(tedx_dataset_path)

tedx_dataset.printSchema()
tedx_dataset = tedx_dataset.filter(length(col('idx')) == 32)
```

```
watch_next_dataset_path = "s3://tedx-data-project/watch_next_dataset.csv"
watch_next_dataset = spark.read.option("header","true").csv(watch_next_dataset_path).dropDuplicates() \
    .filter("url != 'https://www.ted.com/session/new?context=ted.www%2Fwatch-later'")
```

2 – Tedx data collection

Creazione del modello aggregato:

- Ad ogni talk vengono aggregati tutti i relativi tags e gli urls dei watch next consigliati.

```
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") \
```

```
watch_next_dataset_agg = watch_next_dataset.groupBy(col("idx").alias("idx_ref")).agg(collect_list("url").alias("watch_next_urls"))
watch_next_dataset_agg.printSchema()
tedx_dataset_agg = tedx_dataset_agg.join(watch_next_dataset_agg, tedx_dataset_agg._id == watch_next_dataset_agg.idx_ref, "left") \
    .drop("idx_ref") \
    .select(col("*")) \
```

2 – Tedx data collection

Modello aggregato finale:


```
_id: "8cce2e464d0bb2adc24284a827768dd0"
main_speaker: "Simon Sinek"
title: "How to discover your "why" in difficult times"
details: "What has the coronavirus pandemic taught us about ourselves and our re..."
posted: "Posted May 2021"
url: "https://www.ted.com/talks/simon_sinek_how_to_discover_your_why_in_diff..."
num_views: "471,899"
duration: "15:39"
tags: Array
    0: "TED"
    1: "talks"
    2: "leadership"
    3: "mental health"
    4: "self"
    5: "personal growth"
    6: "relationships"
    7: "vulnerability"
watch_next_urls: Array
    0: "https://www.ted.com/talks/scott_dinsmore_how_to_find_work_you_love"
    1: "https://www.ted.com/talks/tracy_young_how_vulnerability_makes_you_a_be..."
    2: "https://www.ted.com/talks/elizabeth_gilbert_it_s_ok_to_feel_overwhelme..."
    3: "https://www.ted.com/talks/heidi_grant_how_to_ask_for_help_and_get_a_ye..."
    4: "https://www.ted.com/talks/lisa_genova_how_your_memory_works_and_why_fo..."
    5: "https://www.ted.com/talks/simon_sinek_why_good_leaders_make_you_feel_s..."
```

3 – Main speaker data collection

Creazione del modello aggregato:

- Ad ogni main speaker presente nel DHW viene aggregata la lista degli urls relativi ai propri talks e, se disponibile, una piccola descrizione recuperata da Wikipedia.

```
urls_dataset_agg = tedx_dataset.groupBy(col("main_speaker").alias("main_speaker_ref")).agg(collect_list("url").alias("related_urls"))
main_speakers_dataset_agg = urls_dataset_agg \
    .join(main_speakers_dataset, urls_dataset_agg.main_speaker_ref == main_speakers_dataset.main_speaker, "left") \
    .select(col("main_speaker").alias("main_speaker_name"), col("main_speaker_description"), col("related_urls"))
```

3 – Main speaker data collection

Modello aggregato finale:


```
_id: ObjectId("60be14deff5a3902b92ebe72")
main_speaker_name: "Adam Grant"
main_speaker_description: "Adam M. Grant (born August 13, 1981) is an American psychologist and a..."
related_urls: Array
    0: "https://www.ted.com/talks/adam_grant_are_you_a_giver_or_a_taker"
    1: "https://www.ted.com/talks/adam_grant_the_surprising_habits_of_original..."
    2: "https://www.ted.com/talks/adam_grant_what_frogs_in_hot_water_can_teach..."
```

4 – Criticità ed evoluzioni

Possibili criticità ed evoluzioni:

- Le criticità maggiori sono causate dal piccolo script Python che recupera le informazioni degli speaker da Wikipedia, infatti:
  - La descrizione restituita non è disponibile per tutti gli speaker.
  - Nessuno ci assicura che la descrizione restituita sia corretta e si riferisca effettivamente all'autore cercato.

- Delle possibili evoluzioni potrebbero essere:
  - Automatizzare il caricamento dei file csv all'interno del Bucket S3.
  - Inglobare lo script python che recupera le informazioni da Wikipedia direttamente all'interno del job pyspark Create Data Lake.
  - Aumentare il numero e la specificità dei tag per ogni talks per permettere una ricerca a granularità fine dei talks.

Grazie