Homework 1 - Big Data Engineering

Anna Lamboglia M63/001219

30 aprile 2022

Indice

1 Traccia

The first homework concerns the definition and implementation of a data model into a NOSQL database about YELP review data on MongoDB or Neo4J (the choice is left to the student as well as how to model data).
In particular, it is required to analyze data available at the following links https://www.yelp.com/dataset in order to define a data model on which the student should performs different queries in order to discuss about the main pros of the defined model. The output of the analysis must be reported within a pdf file of 8/10 pages (whose name will be namesurnamehomework1), whose structure should include the following information:

- Problem introduction and dataset characterization;
- Data model representation;
- Queries and the related output with relevant comments about pros of the proposed model.

This homework can be done individually or in groups of up to two people. In the case of groups, the output must be at least 10 pages.

2 Introduzione al problema e caratterizzazione del Dataset

Il seguente elaborato ha lo scopo di analizzare il dataset fornito da Yelp a questo link: `https://www.yelp.com/dataset`.
Yelp è un social network ed una guida online dove persone, prevalentemente locali, forniscono consigli utili e si scambiano opinioni riguardo a posti ed attività caratteristiche dei luoghi verso cui viaggiano, per lavoro o per vacanza.

Il dataset fornito dalla piattaforma è formato da businesses, recensioni, dati degli utenti, check-in e tips, ed include 6,990,280 recensioni, 150,346 business, 1,987,897 utenti e 12 aree metropolitane.
I file che costituiscono il dataset sono 5 in formato JSON. In particolare:

- *Business.json:* contiene le informazioni dei business registrati a Yelp. I dati includono business id, nome, indirizzo, valutazioni, numero di recensioni, attributi (come ad esempio accesso per sedie a rotelle, cani ammessi, Wi-fi, ecc) che possono variare, categorie, orari di apertura, numero di recensioni ricevute, posizione e così via come mostrato nel codice[1]. La grandezza del file è di 113 Mb;
- *Review.json:* contiene tutte le informazioni relative alle recensioni di un dato business. Sono presenti i campi id recensione, id utente (ossia l'utente che ha scritto la recensione), business id (ossia il business che ha ricevuto la recensione), stelle, data della recensione, contenuto del testo e alcune altre caratteristiche come useful, cool e funny [2]. La grandezza del file è di 4.97 Gb;
- *User.json:* contiene le informazioni sugli utenti registrati a Yelp. Alcuni dei campi di dati includono user id, nome, numero di recensioni, data di iscrizione a Yelp, amici, stelle medie, campi useful, cool, funny e di complimenti come mostrato nel codice [3]. La grandezza del file è di 3.13 Gb;

- *Checkin.json:* contiene i dati di checkin di un business ed è formato semplicemente dal business id e dalla data dei checkin [4]. La grandezza del file è di 273 Mb;
- *Tip.json:* contiene i suggerimenti degli utenti data una certa azienda. I campi presenti sono user id, business id, il testo del suggerimento, la data in cui il suggerimento è stato scritto e e il conteggio dei complimenti [5]. La grandezza del file è di 172 Mb.

```
{'business_id': 'Pns2l4eNsfO8kk83dixA6A',
 'name': 'Abby Rappoport, LAC, CMQ',
 'address': '1616 Chapala St, Ste 2',
 'city': 'Santa Barbara',
 'state': 'CA',
 'postal_code': '93101',
 'latitude': 34.4266787,
 'longitude': -119.7111968,
 'stars': 5.0,
 'review_count': 7,
 'is_open': 0,
 'attributes': {'ByAppointmentOnly': 'True'},
 'categories': 'Doctors, Traditional Chinese Medicine, Naturopathic/Holistic,
    ↪ Acupuncture, Health & Medical, Nutritionists',
 'hours': None}
```

Listing 1: Esempio Business.json

```
{'review_id': 'KU_O5udG6zpxOg-VcAEodg',
 'user_id': 'mh_-eMZ6K5RLWhZyISBhwA',
 'business_id': 'XQfwVwDr-v0ZS3_CbbE5Xw',
 'stars': 3.0,
 'useful': 0,
 'funny': 0,
 'cool': 0,
 'text': "If you decide to eat here, just be aware it is going to take about 2
    ↪ hours from beginning to end. We have tried it multiple times, because I
    ↪ want to like it! I have been to it's other locations in NJ and never had a
    ↪  bad experience. \n\nThe food is good, but it takes a very long time to
    ↪ come out. The waitstaff is very young, but usually pleasant. We have just
    ↪ had too many experiences where we spent way too long waiting. We usually
    ↪ opt for another diner or restaurant on the weekends, in order to be done
    ↪ quicker.",
 'date': '2018-07-07 22:09:11'}
```

Listing 2: Esempio Review.json

```
{'user_id': 'SZDeASXq7o05mMNLshsdIA',
 'name': 'Gwen',
 'review_count': 224,
 'yelping_since': '2005-11-29 04:38:33',
```

```
'useful': 512,
'funny': 330,
'cool': 299,
'elite': '2009,2010,2011',
'friends': 'enx1vVPnfdNUdPho6PH_wg, 4wOcvMLtU6a9Lslggq74Vg, 1
    ↪ OocYCAZixwbAXueW75FMw, GM_iCKAB1eszzcrTZ0zbfg, RgDVC3ZUBqpEe6Y1kPhIpw,
    ↪ i1NwLLny-RDwONN6B6PcWA, PYxHpeTz2ZxZNJEBIAJgvg, iintcYjvi6p0t_6y72KlZg,
    ↪ mBg3-7rnNwLScbARYHI2GQ, ri4ujkuugWObCu9AkLJqAQ, fQl9-OzXtisnX9l9P6A2bQ,
    ↪ bjrgrjBahUQptjfrsGNo7w, io9u-YKgBKdmmivGsynMhQ, S2jpeAzSuQJImtgvRQSmmg,
    ↪ KhjOWuMOTtLuxqqjMaoagA, rhA8QRe-jnRLqR8hBHK1bQ, 7BSP1GSsJkiEXvA3dOPx_Q',
'fans': 28,
'average_stars': 4.27,
'compliment_hot': 24,
'compliment_more': 4,
'compliment_profile': 1,
'compliment_cute': 6,
'compliment_list': 2,
'compliment_note': 12,
'compliment_plain': 16,
'compliment_cool': 26,
'compliment_funny': 26,
'compliment_writer': 10,
'compliment_photos': 9}
```

Listing 3: Esempio User.json

```
{'business_id': '---kPU91CF4Lq2-WlRu9Lw',
 'date': '2020-03-13 21:10:56, 2020-06-02 22:18:06, 2020-07-24 22:42:27,
    ↪ 2020-10-24 21:36:13, 2020-12-09 21:23:33, 2021-01-20 17:34:57, 2021-04-30
    ↪ 21:02:03, 2021-05-25 21:16:54, 2021-08-06 21:08:08, 2021-10-02 15:15:42,
    ↪ 2021-11-11 16:23:50'}
```

Listing 4: Esempio Checkin.json

```
{'user_id': 'NBN4MgHP9D3cw--SnauTkA',
 'business_id': 'QoezRbYQncpRqyrLH6Iqjg',
 'text': 'They have lots of good deserts and tasty cuban sandwiches',
 'date': '2013-02-05 18:35:10',
 'compliment_count': 0}
```

Listing 5: Esempio Tip.json

3 Rappresentazione del modello dei dati

Per portare avanti l'analisi del dataset è stato deciso di utilizzare MongoDB, ossia un DBMS non relazionale sviluppato in C++, open-source, document-oriented e scalabile.
Esso è stato realizzato in maniera tale da avere alte prestazioni, sia in lettura che in scrittura. Tra i vantaggi, si osserva che le letture più consistenti possono essere distribuite in più server replicati e le interrogazioni sono più semplici e veloci grazie all'approccio ai documenti che rende possibile la rappresentazione di relazioni gerarchiche complesse attraverso documenti nidificati e array.

Le caratteristiche principali di MongoDB sono le seguenti:

- *Database document-oriented:* i dati vengono archiviati sotto forma di documenti in formato JSON;
- *Supporto completo agli indici:* indicizzazione di qualsiasi attributo;
- *Replicazione:* facilità nella replicazione dei dati attraverso la rete e alta scalabilità;
- *Sharding:* scalabilità orizzontale senza compromettere nessuna funzionalità;

Avendo chiare le caratteristiche principali, è possibile comprendere le motivazioni che hanno portato alla scelta di questo database NoSQL. In particolare, poiché l'analisi che si vuole effettuare opera con molte operazioni di lettura di file JSON, si necessitava di un database performante e che avesse anche la possibilità di lavorare con degli schemi flessibili e dinamici. Inoltre, è disponibile anche l'integrazione in Python e, in questo modo, è possibile effettuare sia le operazioni di preprocessing che di query in un unico ambiente.

3.1 Preprocessing

Per quanto riguarda la parte di database utilizzato, si è deciso di lavorare principalmente con i seguenti file json:

- Business.json;
- Review.json;

Dato che i file risultano molto pesanti, infatti soltanto il file review è grande circa 4 Gb, si è pensato di limitare il numero di recensioni; dalle 6,990,280 totali, ne sono state analizzate 100,000. Inoltre, l'analisi si è concentrata maggiormente sui business che presentano nel campo *categorie* il tag "Restaurants", ossia i ristoranti, procedendo ad inserire dunque all'interno del database solo quest'ultimi. Stessa cosa è stata fatta con le recensioni, in quanto sono stati preliminarmente presi i businessId dei ristoranti dal file "Business.json" e, successivamente, sono state inserite all'interno del database solo le recensioni relative a quest'ultimi.

Tali operazioni, ed anche le query mostrate nei paragrafi successivi, sono state effettuate attraverso uno script Python che è possibile visionare su Github tramite questo link: `https://github.com/annalamboglia/BigDataHomework`

Ogni file JSON è stato trasformato in una collection all'interno di una istanza di database, mentre i dati del file Json sono diventati dei documenti, tramite i quali è possibile accedere tramite chiave.

4 Query

Per quanto riguarda le query e le proiezioni analizzate, si è deciso di dividerle secondo alcune prospettive, dimostrando anche quanto l'utilizzo di MongoDB possa essere versatile. Le query e le proiezioni sono state suddivise a seconda delle varie prospettive:

- Prospettiva Business;
- Prospettiva Utente;
- Prospettiva Yelp.

4.1 Query e Proiezioni Business

In questo paragrafo sono mostrate le query che un ipotetico business, nel caso in esame un ristorante, potrebbe effettuare.

4.1.1 Query 1

Definire la proiezione per ricavare il numero di recensioni e la media di stelle attuale del business.

```
filter={'name': 'St␣Honore␣Pastries' }
project={ 'name': 1, 'review_count': 1, 'stars':1}

result = collection_business_restaurants.find(filter=filter,projection=project)
result_list = list(result)
result_list
```

Listing 6: Codice Prima Query Business

```
[{'_id': ObjectId('626a74b4d4e365ff14567ece'),
  'name': 'St Honore Pastries',
  'stars': 4.0,
  'review_count': 80}]
```

Figura 1: Risultato Prima Proiezione Business

4.1.2 Query 2

Dato il nome del business, ricavare il testo e le stelle delle recensioni ottenute.

```
pipe = [
    {'$lookup':{
            'from': "yelpBusinessRestaurants",
            'localField': "business_id",
            'foreignField': "business_id",
            'as': "yelpBusiness"
            }
```

```
    }, {'$unwind': "$yelpBusiness"},
    {
        '$match': {"yelpBusiness.name":{'$eq':"Turning␣Point␣of␣North␣Wales"}}
    },
    {'$project': {
            'yelpBusiness.name': 1,
            'text': 1,
            'stars':1}
    }]

result = collection_review_restaurants.aggregate(pipe)
risultato=list(result)
risultato
```

Listing 7: Codice Seconda Query Business

[B]


```
[{'_id': ObjectId('626a78d25fdc03c1ca1cab27'),
  'stars': 4.0,
  'text': "The bun makes the Sonoran Dog. It's like a snuggie for the pup. A first, it seems ridiculous and almost like it's going to be too much, exactly like everyone's favorite
blanket with sleeves. Too much softness, too much smush, too indulgent.  Wrong. It's warm, soft, chewy, fragrant, and it succeeds where other famed Sonoran Dogs fail. \n\nThe hot dog
itself is flavorful, but I would prefer that it or the bacon have a little more bite or snap to better hold their own against the dominant mustard and onions. \n\nI'm with the masses
on the carne asada caramelo. Excellent tortilla, salty, melty cheese, and great carne. \n\nSuper cheap and you can drive through.",
  'yelpBusiness': {'name': 'BK Tacos'}},
 {'_id': ObjectId('626a7aee5fdc03c1ca1cab69'),
  'stars': 4.0,
  'text': "I was told this place is a must for a Sonoran hot dog. I was visiting from out of town and had never had one. It was good, but the whole Sonoran hot dog isn't really my
thing (too many components for me) so I can't say it was great. The guacamole on the other hand, was great! I also had the taco combo - the chicken and carne asada were very good but
the cabeza was slimy. BK has a salsa bar that allows you to help yourself to your own toppings and I appreciated that. Service was good, food arrived quickly. It was pretty packed
during lunch on a weekday.",
  'yelpBusiness': {'name': 'BK Tacos'}}]
```


Figura 2: Risultato Seconda Query Business

4.2 Query e Proiezioni Utente

In questo paragrafo, saranno descritte alcune query che un ipotetico utente potrebbe richiedere.

4.2.1 Query 1

Trovare i ristoranti aperti con più di 4 stelle ed ordinarli in modo decrescente.

```
pipe=[
    {'$match': {
        'stars': {'$gt': 4.0},
        'is_open':{'$eq':1}}},
    {'$sort' : { 'stars' : -1, 'city': 1 } },
    {'$project': {'_id': 0,'name': 1,'city':1,'stars':1,'is_open':1}}]

result = collection_business_restaurants.aggregate(pipe)
result_list = list(result)
```

```
result_list
```

Listing 8: Codice Prima Query Utente

```
[{'name': '2 Fat Dogs', 'city': 'Abington', 'stars': 5.0, 'is_open': 1},
 {'name': "Long John Silver's", 'city': 'Arnold', 'stars': 5.0, 'is_open': 1},
 {'name': 'Q Tea Vietnamese Cafe',
  'city': 'Arnold',
  'stars': 5.0,
  'is_open': 1},
 {'name': "Ryan's Country Store", 'city': 'Aston', 'stars': 5.0, 'is_open': 1},
 {'name': 'Desserts by Design', 'city': 'Audubon', 'stars': 5.0, 'is_open': 1},
 {'name': 'Core de Roma', 'city': 'Bala Cynwyd', 'stars': 5.0, 'is_open': 1},
 {'name': 'Rosalia's Pizza', 'city': 'Berlin', 'stars': 5.0, 'is_open': 1},
 {'name': 'Graveyard B&G', 'city': 'Boise', 'stars': 5.0, 'is_open': 1},
 {'name': 'ā café', 'city': 'Boise', 'stars': 5.0, 'is_open': 1},
 {'name': 'Off the Grid Pizza', 'city': 'Boise', 'stars': 5.0, 'is_open': 1}]
```

Figura 3: Risultato Prima Query Business

4.2.2 Query 2

Trovare i ristoranti aperti in una certa città ordinandoli per stelle.

```
pipe=[{'$match': {'stars': {'$gt': 4.0},'is_open':{'$eq':1}}},
    {'$sort' : { 'stars' : -1, 'city': 1 } },
    {'$project': {'_id': 0,'name': 1,'city':1,'stars':1,'is_open':1}
    }]

result = collection_business_restaurants.aggregate(pipe)
result_list = list(result)
result_list
```

Listing 9: Codice Seconda Query Utente

```
[{'name': 'El Guero  Mexican Food Truck',
  'address': '1256 W Montgomery Ave',
  'stars': 5.0,
  'is_open': 1,
  'attributes': {'RestaurantsTakeOut': 'True',
   'OutdoorSeating': 'False',
   'RestaurantsTableService': 'False',
   'Caters': 'True',
   'HasTV': 'False',
   'RestaurantsDelivery': 'True',
   'Alcohol': "u'none'",
   'WiFi': "u'no'",
   'WheelchairAccessible': 'True',
   'BusinessAcceptsCreditCards': 'True',
   'BikeParking': 'True'},
  'hours': {'Monday': '10:0-18:0',
   'Tuesday': '10:0-18:0',
   'Wednesday': '10:0-18:0',
   'Thursday': '10:0-18:0',
   'Friday': '10:0-18:0'}},
 {'name': 'Jade Palace',
  'address': '1714 S 5th St',
  'stars': 5.0,
  'is_open': 1,
  'attributes': {'HasTV': 'True', 'RestaurantsDelivery': 'True'},
```


Figura 4: Risultato Seconda Query Cliente

4.3 Query e Proiezioni Yelp

In questo paragrafo sono analizzate le query che Yelp potrebbe effettuare per valutare l'andamento dei business e delle review all'interno della piattaforma.

4.4 Query 1

Mostare le prime 10 città che presentano più ristoranti.

```
pipe=[{'$group':{'_id': '$city', 'Totale':{ '$count': {}}}},
    {'$sort': {'Totale':-1}},
    {'$limit' : 10}]

result=collection_business_restaurants.aggregate(pipe)
res = list(result)
res
```

Listing 10: Codice Prima Query Yelp

```
[{'_id': 'Philadelphia', 'Totale': 1730},
 {'_id': 'Tampa', 'Totale': 848},
 {'_id': 'Indianapolis', 'Totale': 827},
 {'_id': 'Tucson', 'Totale': 742},
 {'_id': 'Edmonton', 'Totale': 693},
 {'_id': 'Nashville', 'Totale': 660},
 {'_id': 'New Orleans', 'Totale': 602},
 {'_id': 'Saint Louis', 'Totale': 528},
 {'_id': 'Reno', 'Totale': 366},
 {'_id': 'Boise', 'Totale': 230}]
```

Figura 5: Risultato Prima Query Yelp

4.5 Query 2

Mostrare i ristoranti aperti con il maggior numero di recensioni.

```
filter={
   'is_open': {
       '$ne': 0
    }
}
project={
    'name':1,
    'review_count': 1,
     'stars': 1,
    '_id': 0
}
sort=list({
    'review_count': -1,
    'stars':-1
}.items())
limit = 100
result = collection_business_restaurants.find(
    filter=filter,
    projection=project,
    sort = sort,
    limit = limit
)
result_list = list(result)
result_list
```

Listing 11: Codice Seconda Query Yelp

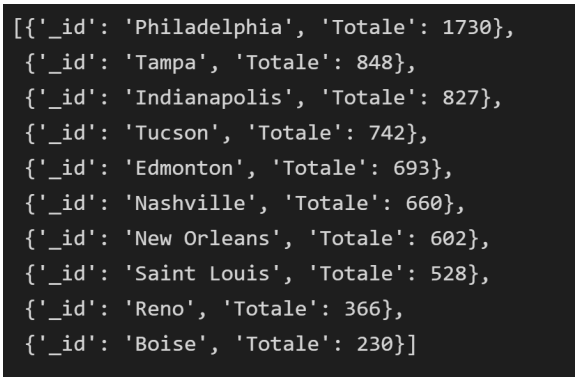

Figura 6: Risultato Prima Query Yelp

4.6 Query 3

Con MongoDB è possibile adottare tecniche semplici per effettuare Sentimental Analysis, in quanto è possibile ricercare i documenti che presentano all'interno determinati caratteri o parole. Ad esempio, qualora si definisse in partenza una "Bag of Words" contenente le parole che identificano una recensione positiva, potrebbe essere interessante eseguire delle query per ottenere il numero di recensioni positive all'interno della piattaforma. Lo stesso approccio può essere eseguito nella valutazione di recensioni negative. Di seguito si riporta un esempio di come può essere creato un semplice filtraggio tramite alcune parole chiave.

```
filter={
    '$or' : [
        { 'text': {'$regex': re.compile(r"good")}},
        { 'text': {'$regex': re.compile(r"amazing")}},
        { 'text': {'$regex': re.compile(r"great")}},
        { 'text': {'$regex': re.compile(r"cool")}}
    ]
}

project={ 'name': 1, 'stars': 1, 'text':1, '_id':0}

result = collection_review.find(filter=filter,projection=project)

result_list = list(result)
result_list
```

Listing 12: Codice Terza Query Yelp

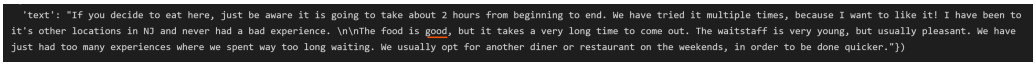

Figura 7: Risultato Terza Query Yelp

5 Indici

Per migliorare le prestazioni per la ricerca dei documenti di cui si necessita è possibile utilizzare degli indici. In particolare, nel caso in esame potrebbe essere interessante inserire un indice relativo al campo "city" in quanto è spesso utilizzato all'interno delle query per ricercare determinati ristoranti in una certa città. La tecnica degli indici è molto utile, anche se è necessario comprendere al meglio quando utilizzarla, in quanto, è vero che permette di velocizzare la ricerca, d'altro canto gli indici occupano spazio in memoria, dunque bisogna trovare il giusto trade off. Per poter creare un indice è stata eseguita la seguente riga di codice:

```
db.collection_business_restaurants.create_index([ ("city", ASCENDING) ])
```

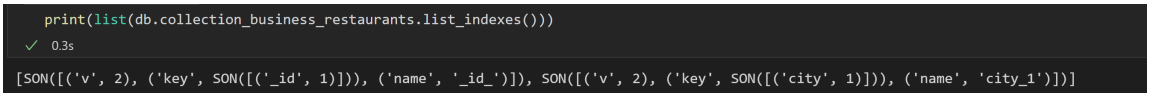

Figura 8: Indici