Scopo dell’analisi: effettuare una sentiment analisys in modo da attribuire un punteggio alle vare review. Le review devono essere opportunamente pesate anche in relazione all’utente che le ha espresse.

DOMANDE IMPORTANTI:

1. Come vogliamo pesare gli utenti e le review? Ci serve una formula!
2. Dobbiamo fare l’analisi con librerie già fatte oppure costruire un sistema che si basa sul classificatore e quindi predire la positività delle review!!! Importante

* GIORNO 14/06/2018
  + Funzionamento mongoDB e input dei file nel database
  + Modifica della variabile path per mettere il percorso di spark per chiamare la shell più velocemente
  + Installazione delle librerie di python (nltk)
  + Analisi del file user.json:

Nel db i file sono stati inseriti nel seguente modo:

mongoimport --db test --collection docs --file example2.json

Comandi mongodb

* Start: sudo service mongod start
* Stop: come sopra solo stop
* Mongosb shell: mongo --host 127.0.0.1:27017
  + Use YelpDB
* db.createCollection("user")
* db.createCollection("review")
* db.createCollection("tip")
* db.createCollection("business")
* show collections
* ctrl c
* mongoimport --db YelpDB --collection user --file user.json

ESEMPIO DI UTENTE

{

"\_id" : ObjectId("5b22a02518db4a80f9916701"),

"user\_id" : "h5\_D5TlEN4bREoy3vR-Vxw", ID UTENTE

"name" : "John", NOME

"**review\_count**" : 1, CONTEGGIO DI REVIEW

"yelping\_since" : "2017-10-25", DATA DI ISCRIZIONE ALLA PIATTAFORMA

"**friends**" : [ ], LISTA DI AMICI

"**useful**" : 0, VOTI USEFUL MESSI

"**funny**" : 0, VOTI FUNNY MESSI

"**cool**" : 0, NUMERI DI VOTI COOL

"**fans**" : 0, NUMERO DI FAN

"**elite**" : [ ], LISTA DI ANNI IN CUI UTENTE ELITE

"average\_stars" : 1, MEDIA DI VOTI DATI DALL’UTENTE

"**compliment\_hot**" : 0, COMPLIMENTI DA URLO RICEVUTI

"**compliment\_more**" : 0, “SCRIVI DI Più” RICEVUTI

"**compliment\_profile**" : 0, BEL PROFILO

"**compliment\_cute**" : 0, BELLA FOTO sei carico

"**compliment\_list**" : 0, ELENCHI FANTASTICI

"compliment\_note" : 0, complimento just a note cioè ti voglio dire giusto una cosa in merito a una tua foto o commendo

"compliment\_plain" : 0, GRAZIE RICEVUTI

"compliment\_cool" : 0, SEI FIGO RICEVUTI

"compliment\_funny" : 0, SEI DIVERTENTE RICEVUTI

"compliment\_writer" : 0, SCRIVI BENE RICEVUTI

"compliment\_photos" : 0 BELLE FOTO RICEVUTI

}

(s=3.7)

**ValoreUtente: 0.27\*Elite18+ numAiElite0.05+0.13fan+0.06reviewcount+0.1namici+0.1sCompl+ 0.19useful+0.03funny+0.05cool**

La prima cosa che è stata fatta è vedere se ci sono utenti che non hanno mai fatto una review, quindi abbiamo fatto la **query db.user.find({“review\_count” : 0}).count().** La query ha ritornato 1272 persone che non hanno fatto review. In effetti dobbiamo analizzare le review e quindi non ha senso avere utenti che non hanno mai fatto review.

Eliminiamo gli utenti che non hanno mai fatto review con il comando: **db.user.deleteMany({“review\_count” : 0}).**

Adessi si analizza il file review.json

{

"\_id" : ObjectId("5b22a0fd18db4a80f9a5a859"),

"review\_id" : "K7o5jDInfmX3cY5oH6ATNw", ID

"user\_id" : "u0LXt3Uea\_GidxRW1xcsfg", utente che ha fatto la recensione

"business\_id" : "oWTn2IzrprsRkPfULtjZtQ", id del negozio

"stars" : 3, stelle associate

"date" : "2012-09-23", data della recensione

"text" : recensione test

"useful" : 4, quanti voti useful ha avuto

"funny" : 0, quanti voti funny ha avuto

"cool" : 0 quanti voti cool ha avuto

}

Normalizzazione:

**ValoreReview: (lib+sterlinenormalizzato)\*(****1+useful0.7+funny0.1+cool0.2 ) \*(1+ valoreUtente)**

Ci sono 5261669 review che dobbiamo analizzare. (per saperlo abbiamo fatto **db.review.find().count())**

* GIORNO 15/06/18

Analisi del file business.json

Questo sotto è solo un esempio perché i campi variando in relazione al tipo di negozio.

{ "\_id" : ObjectId("5b22a76e0cf13e097ce91e3b"),  
"business\_id" : "VSGcuYDV3q-AAZ9ZPq4fBQ",  
"name" : "Sportster's",  
 "neighborhood" : "The Danforth",  
"address" : "1430 Danforth Avenue",  
"city" : "Toronto",  
"state" : "ON",  
"postal\_code" : "M4J 1N4",  
"latitude" : 43.6828673,  
"longitude" : -79.3269639,  
"stars" : 2.5,  
"review\_count" : 7,  
"is\_open" : 1,  
"attributes" : {  
"BusinessParking" : {  
"garage" : false,  
"street" : false,  
"validated" : false,  
"lot" : false,  
"valet" : false},  
"HasTV" : true,  
"CoatCheck" : false,  
"OutdoorSeating" : true,  
"BusinessAcceptsCreditCards" : true,  
"RestaurantsPriceRange2" : 3,  
"WheelchairAccessible" : true,  
"BikeParking" : true,  
 "Smoking" : "outdoor",  
"BestNights" : {  
"monday" : false,  
"tuesday" : false,  
"friday" : true,  
"wednesday" : false,  
"thursday" : false,  
"sunday" : false,  
"saturday" : true},  
"RestaurantsGoodForGroups" : true,  
"NoiseLevel" : "average",  
"GoodForDancing" : false,  
"Alcohol" : "full\_bar"  
},  
"categories" : [  
"Bars",  
"Sports Bars",  
"Nightlife"],

"hours" : {

"Monday" : "16:00-2:00",

"Tuesday" : "16:00-2:00",

"Friday" : "16:00-2:00",

"Wednesday" : "16:00-2:00",

"Thursday" : "16:00-2:00",

"Sunday" : "16:00-2:00",

"Saturday" : "16:00-2:00"

}

}

Tutti i negozi hanno delle review e quindi non possiamo eliminare nulla. (campo rview\_count)

* + Analisi di tip.json

{

"\_id" : ObjectId("5b22a7150cf13e097cd85792"),

"text" : "Hot here too. Dang!",

"date" : "2012-06-20",

"likes" : 0,

"business\_id" : "yQab5dxZzgBLTEHCw9V7\_w",

"user\_id" : "TvkqJ8YEIsTb16RnnrNyfQ"

}

<https://pip.pypa.io/en/stable/installing/> PER ISTALLARE PIP

<https://planspace.org/20150607-textblob_sentiment/>

Script da implementare:

1. Analisi e calcolo del valore utente. In uscita un file json con nome sintesiuser.json così formato:  
   { user\_id  
   valore}
2. Calcolo valore review. In uscita un file json con nome sintesireview.json così formato  
   { user\_id  
   review\_id  
   business\_id  
   data  
   valorerevi  
   valoreoppublica}
3. Uno script che riporti le parole più frequenti.

Calcolo valore delle review: Assunzioni:

* Prendiamo in considerazione dei business che hanno un numero di review giusto per poter effettuare una analisi (500)

Calcolo valore Utente :

* Nel db gli utenti che non ci servono sono stati già eliminati. Quindi devi lavorare su tutto quello che sta nel db.
* Nello script devi caricare gli utenti. Scorrere tutto il cursore che ti ritorna per calcolare i valori.
* Crea quindi un altro json. Lo salvi sul file e poi devi trovare su internet come caricare il file sul db YelpUser in una nuova collection (chiamala come chiami il file)
* Non dovresti avere problemi. Vedi come ho fatto io nello script valorereview.
* L’utente generato deve avere due capi: id e valore.

**ValoreUtente: 0.27\*Elite17+ numAiElite0.05+0.13fan+0.08reviewcount+0.08namici+0.1sCompl+ 0.19useful+0.03funny+0.05cool**

* GIORNO 24/06/218

Valore review -> ok

Valore Utente -> ok.

Interpretazione valore review:

La bontà della review è data da. Lib+0.5\*stars-1.5. Lib è calcolata facendo una media della polarità di ogni frase e quindi è compresa tra -1 e 1. Stars invece è costituito da 1 2 3 4 5. Anche in questo caso il valore minimo è -1 e il massimo è 1. Allora la bontà di una review va da -2 a 2. In particolare se il valore è positivo parliamo di review positiva, altrimenti negativa.

Mi serve :

* Max valoreUtente
* Max useful
* Max funny
* Max cool

Per stabilire il valore massimo dell’opinione su un negozio.

Analisi da fare:

1. Percentuale review positive/negative
2. Andamento temporale opinione pubblica.
3. Confronto tra le stelle e l’analisi della review. (quanto coincidono ??)
4. Parole più usate per il negozio.

-ordine di girare script

1. Valore utente -> sintesiuser
2. Valore review -> sintesireviewidea
3. Tuning dei parametri
4. Valore review ottimizzato -> sintesireviewopt
5. Filtraggio
6. Calcolo dei valori cumulativi
7. Frequenza parole