CIS5200 Term Project Tutorial

Authors: Yougender Chauhan, Nagender Chauhan, Suman Chauhan, Vishwanth Reddy Sama.

Instructor: Jongwook Woo

Date: 12/16/2022

Lab Tutorial

Yougender Chauhan (ychauha4@calstatela.edu)

Nagender Chauhan (nchauha5@calstatela.edu)

Suman Chauhan (schauha7@calstatela.edu)

Vishwanth Reddy Sama (vsama@calstatela.edu)

12/16/2022

Sentiment Analysis on Amazon Book Review Dataset

Objectives

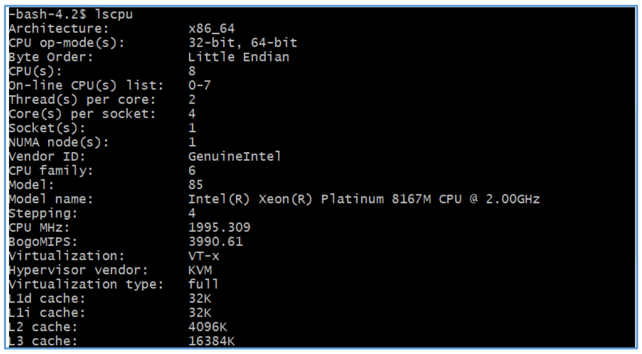
In this hands-on lab, you will learn how to:

- Download Amazon book review data with -wget command.
- Upload the data into HDFS using -put command.
- Create Hive tables in HDFS using Hive/beeline QL
- Analysis on data using HIVE/beeline QL.
- Using a pre-determined Dictionary to do sentiment analysis on books reviews and their summary.

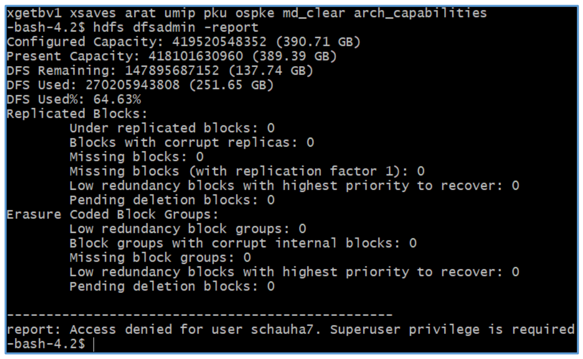
Platform Spec

- Cluster Version: Hadoop 3.1.2
- CPU Speed: 1995.309
- # of CPU cores: 4
- # of nodes: 5 (2 – Master and 3 – Worker)
- Total Memory Size: 390.71 GB

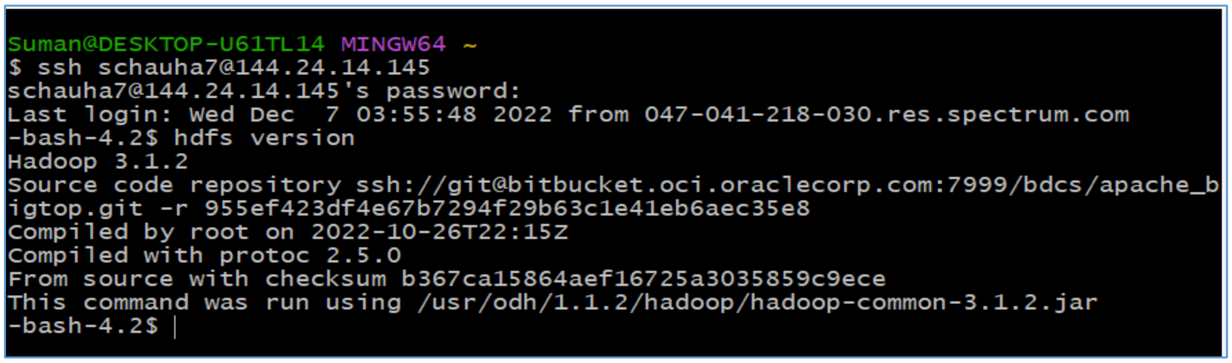
$ hdfs version
$ lscpu
$ hdfs dfsadmin -report



```
-bash-4.2$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                8
On-line CPU(s) list:   0-7
Thread(s) per core:    2
Core(s) per socket:    4
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 85
Model name:            Intel(R) Xeon(R) Platinum 8167M CPU @ 2.00GHz
Stepping:              4
CPU MHz:               1995.309
BogoMIPS:              3990.61
Virtualization:        VT-x
Hypervisor vendor:     KVM
Virtualization type:   full
L1d cache:             32K
L1i cache:             32K
L2 cache:              4096K
L3 cache:              16384K
```



```
xgetbv1 xsaves arat umip pku ospke md_clear arch_capabilities
-bash-4.2$ hdfs dfsadmin -report
Configured Capacity: 419520548352 (390.71 GB)
Present Capacity: 418101630960 (389.39 GB)
DFS Remaining: 147895687152 (137.74 GB)
DFS Used: 270205943808 (251.65 GB)
DFS Used%: 64.63%
Replicated Blocks:
        Under replicated blocks: 0
        Blocks with corrupt replicas: 0
        Missing blocks: 0
        Missing blocks (with replication factor 1): 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0
Erasure Coded Block Groups:
        Low redundancy block groups: 0
        Block groups with corrupt internal blocks: 0
        Missing block groups: 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0

-------------------------------------------------
report: Access denied for user schauha7. Superuser privilege is required
-bash-4.2$ |
```



```
Suman@DESKTOP-U61TL14 MINGW64 ~
$ ssh schauha7@144.24.14.145
schauha7@144.24.14.145's password:
Last login: Wed Dec  7 03:55:48 2022 from 047-041-218-030.res.spectrum.com
-bash-4.2$ hdfs version
Hadoop 3.1.2
Source code repository ssh://git@bitbucket.oci.oraclecorp.com:7999/bdcs/apache_b
igtop.git -r 955ef423df4e67b7294f29b63c1e41eb6aec35e8
Compiled by root on 2022-10-26T22:15Z
Compiled with protoc 2.5.0
From source with checksum b367ca15864aef16725a3035859c9ece
This command was run using /usr/odh/1.1.2/hadoop/hadoop-common-3.1.2.jar
-bash-4.2$ |
```

Step 1: Downloading and uploading dataset into HDFS

Explain what this step is for. This step is to download the amazon book review dataset which is at my GitHub repository into the /tmp Linux server (since the file size is about 3gb) using wget command in gitbash CLI, then unzipping the dataset, uploading it into the HDFS.

1. Open Getbash CLI, connect to your Linux server using $ ssh username@ipaddress.

```
$ cd /tmp

$ wget https://github.com/Chauhan67/Amazonbookreviews/releases/download/v1/amazonbooks.zip

$ unzip amazonbooks.zip -d /tmp/books

$ cd /books

$ ls

$ hdfs dfs -mkdir books_data

$ hdfs dfs -mkdir Books_rating

$ hdfs dfs -put books_data.csv books_data

$ hdfs dfs -put Books_rating.csv Books_rating

$ cd
```

2. Run the above codes in this order to first get into /tmp directory to download .zip dataset, after which unzip it into /tmp/books sub-directory, create two directories in hdfs using hdfs dfs -mkdir command, upload into hdfs using -put command. Then use 'cd' in bash to go to your home directory i.e /home/<your-username>

```
$ wget -O dictionary.tsv https://github.com/dalgual/aidatasci/raw/master/data/bigdata/dictionary.tsv

$ hdfs dfs -mkdir dictionary

$ hdfs dfs -put dictionary.tsv dictionary
```

3. Now, download the dictionary in your Linux home directory, upload it into HDFS and view all files using ls and cat command.

Then you will see the following : `-bash-4.2$ hdfs dfs -ls`

```
$ hdfs dfs -ls

$ hdfs dfs -cat books_data/books_data.csv | head -n 10

$ hdfs dfs -cat Books_rating/Books_rating.csv | tail -n 10
```

```
Found 6 items
drwx----w-   - ychauha4 hdfs          0 2022-12-07 05:05 .Trash
drwxr-xrwx   - ychauha4 hdfs          0 2022-11-10 02:10 .hiveJars
drwxr-xr-x   - ychauha4 hdfs          0 2022-12-03 00:07 Books_rating
drwxr-xr-x   - ychauha4 hdfs          0 2022-12-03 00:07 books_data
drwxr-xr-x   - ychauha4 hdfs          0 2022-12-03 01:08 dictionary
drwxr-xr-x   - ychauha4 hdfs          0 2022-12-07 03:48 tmp
```

For -cat commands : `-bash-4.2$  hdfs dfs -cat books_data/books_data.csv | head -n 10`

```
Title,description,authors,image,previewLink,publisher,publishedDate,infoLink,categorie
s,ratingsCount

Its        Only        Art        If        Its        Well        Hung!,,['Julie
Strain'],http://books.google.com/books/content?id=DykPAAAACAAJ&printsec=frontcover&img
=1&zoom=1&source=gbs_api,http://books.google.nl/books?id=DykPAAAACAAJ&dq=Its+Only+Art+
If+Its+Well+Hung!&hl=&cd=1&source=gbs_api,,1996,http://books.google.nl/books?id=DykPAA
AACAAJ&dq=Its+Only+Art+If+Its+Well+Hung!&hl=&source=gbs_api,['Comics & Graphic Novels'],
```

```
-bash-4.2$ hdfs dfs -cat Books_rating/Books_rating.csv | tail -n 10

B000NSLVCU,The    Idea    of    History,,AI1QNMVF2E3TN,"Robin    George    ""Master    of
Arts""",28/29,5.0,1057017600,R.    G.    Collingwood's    Most    Famous    Book,"Highly
Recommended.This book is one of the best books ever written on the Nature and Aims of
History. This along with his &quot;Principles of History&quot; should give most readers
all they need to know about the how and why of history.The book is extremely easy to
read; harder to understand. Some criticisms of the book are not up to the mark, as for
example complaints that Collingwood used Greek and Latin phrases in the book, and not
everyone understands them.
```

Step 2: Data and sentiment analysis in Hive/beeline

Here, we will use hive or in our case beeline CLI to create tables and query data for analysis. We will be creating External Tables, Views, N-gram, Explode() etc for querying and analysis of the dataset.

```
-bash-4.2$ beeline
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> use <yourdatabase-name>;
```

You'll see the following : `Driver: Hive JDBC (version 3.1.2)`

```
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.2 by Apache Hive
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai>
```

1. Creating External Tables for books_data, Books_rating and dictionary by using out datasets. To do this use the following code:

```
CREATE EXTERNAL TABLE IF NOT EXISTS books_data (title STRING,
        description STRING,
        authors STRING,
        image STRING,
        preview STRING,
        publisher STRING,
        publish_date BIGINT,
        info_link STRING,
        categories STRING,
        ratings_count float)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/ychauha4/books_data'
TBLPROPERTIES ('skip.header.line.count'='1');
```

NOTE – Replace ychauha4 with your username in the above code, after creating the table, to check we run the following select code:

```
select * from books_data limit 3;
```

You will see the following result with the column name – but not the header listed in the csv file:

```
+----------------------------------------+--------------------------------------------
--------+----------------------------------------------------+------------------------
---------------------------------+----------------------------------------------------
+--------------------------+----------------------------------------------------+-----
----------------------------------
|             books_data.title             |                books_data.description
|                  books_data.authors                  |
books_data.image                          |                books_data.publisher
| books_data.publish_date  |                books_data.info_link                |
books_data.categories
+----------------------------------------+--------------------------------------------
--------+----------------------------------------------------+------------------------
---------------------------------+----------------------------------------------------
+--------------------------+----------------------------------------------------+-----
----------------------------------
| Its Only Art If Its Well Hung!          |
| ['Julie Strain']                                     |
http://books.google.com/books/content?idource=gbs_api |
http://books.google.nl/books?id=DykPAAAACAAJ&dq=Its+Only+Art+If+Its+Well+Hung!&hl=&cd=
1&source=gbs_api |                                                  | 1996
AAAACAAJ&dq=Its+Only+Art+If+Its+Well+Hung!&hl=&source=gbs_api | ['Comics & Graphic
Novels']                        | NULL                        |
| Dr. Seuss: American Icon              | "Philip Nel takes a fascinating look into
the key aspects of Seuss's career - his poetry |  politics
|  a |  marketing                                        |  and place in the popular
imagination."" ""Nel argues convincingly that Dr. Seuss is one of the most influential
poets in America. Hias changed language itself                      |  giving us new
words like ""nerd."" And Seuss's famously loopy artistic style - what Nel terms an
""energetic cartoon surrealism"" - has b  |
| Wonderful Worship in Smaller Churches  | "This resource includes twelve principles
in understanding small church worship |  fifteen practices for planning worship with
fewer than 100 peop        | ['David R. Ray']                                  |
http://books.google.com/books/content?id=2tsDAAAACAAJ&printsec=frontcover&img=1&zoom=1
&source=gbs_api | NULL                          | 2000
| NULL                     |
+-----------------------------------------+-------------------------------------------
--------+------------------------------------------------+------------------------
---------------------------+---------------------------------------------------------
+-------------------------+----------------------------------------------------+-----
----------------------------------
3 rows selected (0.333 seconds)
```

Create another external table for the books rating which has reviews for each book.

```
CREATE EXTERNAL TABLE IF NOT EXISTS books_rating (id BIGINT,
        title STRING,
        price FLOAT,
        user_id STRING,
        profile_name STRING,
        r_helpfulness STRING,
        r_score INT,
        r_time BIGINT,
        r_summary STRING,
        r_review STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/ychauha4/Books_rating'
TBLPROPERTIES ('skip.header.line.count'='1');
```

NOTE – Replace ychauha4 with your username in the above code, after creating the table, to check we run the following select code:

```
select * from books_rating limit 3;
```

You will see the following result with the column name – but not the header listed in the csv file:

```
+-------------------+----------------------------------+---------------------+----------
-------------+----------------------------+-----------------------------+-------------
----------+----------------------+-------------------------------------------------+-
---------------------------------------------------+
| books_rating.id  |       books_rating.title       | books_rating.price  |
books_rating.user_id  | books_rating.profile_name  | books_rating.r_helpfulness  |
books_rating.r_score  | books_rating.r_time  |            books_rating.r_summary
|              books_rating.r_review              |
+------------------+--------------------------------+---------------------+----------
-------------+---------------------------+-----------------------------+------------
----------+----------------------+---------------------------------------------------+-
---------------------------------------------------+
| 1882931173       | Its Only Art If Its Well Hung!  |                     |
AVCGYZL8FQQTD         | "Jim of Oz ""jim-of-oz"""  | 7/7                         | 4
| 940636800           | Nice collection of Julie Strain images          | "This is
only for Julie Strain fans. It's a collection of her photos -- about 80 pages worth
with a nice section of paintings by Olivia.If you're looking for heavy literary
content |
| 826414346        | Dr. Seuss: American Icon        |                     |
A30TK6U7DNS82R        | Kevin Killian              | 10/10                       | 5
| 1095724800          | Really Enjoyed It                               | "I don't
care much for Dr. Seuss but after reading Philip Nel's book I changed my mind--that's
a good testimonial to the power of Rel's writing and thinking. Rel plays Dr. Seuss the
ultimate compliment of treating him as a serious poet as well as one of the 20th
century's most interesting visual artists |
| 826414346        | Dr. Seuss: American Icon        |                     |
A3UH4UZ4RSVO82        | John Granger               | 10/11                       | 5
| 1078790400          | Essential for every personal and Public Library  | "If people
become the books they read and if ""the child is father to the man |
+------------------+--------------------------------+---------------------+----------
-------------+---------------------------+-----------------------------+------------
----------+----------------------+---------------------------------------------------+-
---------------------------------------------------+
3 rows selected (0.384 seconds)
```

Now Create a table for our dictionary which we will use for our sentiment analysis .

```
CREATE EXTERNAL TABLE if not exists dictionary (
        type string,
        length int,
        word string,
        pos string,
        stemmed string,
        polarity string )
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION '/user/ychauha4/dictionary';
```

NOTE – Replace ychauha4 with your username in the above code, after creating the table, to check we run the following select code:

```
select * from dictionary limit 3;
```

```
+------------------+--------------------+------------------+-----------------+--------
-------------+---------------------+
| dictionary.type  | dictionary.length  | dictionary.word  | dictionary.pos  |
dictionary.stemmed  | dictionary.polarity  |
+------------------+--------------------+------------------+-----------------+--------
------------+---------------------+
| weaksubj         | 1                  | abandoned        | adj             | n
| negative             |
| weaksubj         | 1                  | abandonment      | noun            | n
| negative             |
| weaksubj         | 1                  | abandon          | verb            | y
| negative             |
+------------------+--------------------+------------------+-----------------+--------
-------------+---------------------+
3 rows selected (0.395 seconds)
```

Now, to see the number of records for books_data and books_rating tables we have created, use the following code:

```
select count(*) from books_data;

select count(*) from Books_rating;
```

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> select count(*) from books_data;
+---------+
|   _c0   |
+---------+
| 212404  |
+---------+
1 row selected (13.111 seconds)
```

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> select count(*) from Books_rating;
+----------+
|   _c0    |
+----------+
| 3000000  |
+----------+
1 row selected (12.699 seconds)
```

From this we know that we have 212404 different books and 3Million book reviews for the 212404 books.

2. Now, for the analysis we create a view to with title, average score and count of number of reviews for each book, we query this view to get most reviewed and least rated for more than 500 reviews then we use INSERTOVER DIRECTORY to save into our HDFS.

```
CREATE VIEW IF NOT EXISTS top10books AS

select title, avg(r_score) as avg_score, count(*) as count from books_rating group by title;
```

Next, to get top 10 rated books, their average rating and ordered by the count of reviews run this code:

```
INSERT OVERWRITE DIRECTORY '/user/ychauha4/tmp/top10'

ROW FORMAT DELIMITED FIELDS TERMINATED BY','

select * from top10books sort by count DESC limit 10;
```

NOTE – Change ychauha4 with your username.

You'll get the following output saved to /tmp/top10/000000_0;

```
----------------------------------------------------------------------------------------
--------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED
KILLED
----------------------------------------------------------------------------------------
--------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0
0
Reducer 2 ...... container     SUCCEEDED     22         22        0        0       0
0
Reducer 3 ...... container     SUCCEEDED     11         11        0        0       0
0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0
0
----------------------------------------------------------------------------------------
--------
VERTICES: 04/04  [==========================>>] 100%  ELAPSED TIME: 38.35 s
----------------------------------------------------------------------------------------
--------
No rows affected (44.282 seconds)
```

Now to get least 10 reviewed books with at least 500 reviews to it.

```
INSERT OVERWRITE DIRECTORY '/user/ychauha4/tmp/least10'

ROW FORMAT DELIMITED FIELDS TERMINATED BY','

select * from top10books where avg_score IS NOT NULL AND count >=500 sort by avg_score limit 10;
```

NOTE – Change ychauha4 with your username.

You'll get the following output saved to /tmp/least10/000000_0;

```
----------------------------------------------------------------------------------------
--------
```

```
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED
KILLED
----------------------------------------------------------------------------------------
--------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0
0
Reducer 2 ...... container     SUCCEEDED     22         22        0        0       0
0
Reducer 3 ...... container     SUCCEEDED      3          3        0        0       0
0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0
0
----------------------------------------------------------------------------------------
--------
VERTICES: 04/04  [==========================>>] 100%  ELAPSED TIME: 37.76 s
----------------------------------------------------------------------------------------
--------
No rows affected (43.226 seconds)
```

Now for top10categories by count run the following code:

```
INSERT OVERWRITE DIRECTORY '/user/ychauha4/tmp/top10categories'

ROW FORMAT DELIMITED FIELDS TERMINATED BY','

select categories, count(categories) as count from books_data where categories RLIKE
'(?<=\').+?(?=\')' group by categories sort by count DESC limit 10;
```

NOTE – Change ychauha4 with your username.

You'll get the following output saved to /tmp/top10categories/000000_0;

```
----------------------------------------------------------------------------------------
--------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED
KILLED
----------------------------------------------------------------------------------------
--------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0
0
Reducer 2 ...... container     SUCCEEDED      6          6        0        0       0
0
Reducer 3 ...... container     SUCCEEDED      4          4        0        0       0
0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0
0
----------------------------------------------------------------------------------------
--------
VERTICES: 04/04  [==========================>>] 100%  ELAPSED TIME: 18.56 s
----------------------------------------------------------------------------------------
--------
No rows affected (23.865 seconds)
```

To check if the files have been downloaded to HDFS, in another gitbash window with your Linux system, run the below command:

```
$ hdfs dfs -ls tmp

$ hdfs dfs -cat tmp/top10/000000_0 |head -n 3

$ hdfs dfs -cat tmp/least10/000000_0 |head -n 3

$ hdfs dfs -cat tmp/top10categories/000000_0 |head -n 3
```

You should see the following outputs:

```
-bash-4.2$ hdfs dfs -ls tmp
Found 2 items
drwxr-xr-x   - ychauha4 hdfs          0 2022-12-17 07:26 tmp/least10
drwxr-xr-x   - ychauha4 hdfs          0 2022-12-17 07:19 tmp/top10
```

```
-bash-4.2$ hdfs dfs -cat tmp/top10/000000_0 |head -n 3
The Hobbit,4.655425755224971,22023
Pride and Prejudice,4.527145185990219,20371
Atlas Shrugged,4.026377349358715,12513
```

```
-bash-4.2$ hdfs dfs -cat tmp/least10/000000_0 |head -n 3
Red Rabbit,2.1573208722741435,643
Killing Time,2.3694029850746268,538
The Bear and the Dragon,2.4647627821280516,2179
```

```
-bash-4.2$ hdfs dfs -cat tmp/top10categories/000000_0 |head -n 3
['Fiction'],2123
['Religion'],1250
['History'],931
```

We will download this later with other files for visualization.

3. N-gram and context n-gram analysis of the most reviewed book i.e. 'The Hobbit' and least rated book 'Red Rabbit' first for 'The Hobbit' let us do a trigram. Run the below code:

```
SELECT EXPLODE(context_ngrams(sentences(LOWER(r_review)),

array(null, null, null), 10)) AS trigram FROM books_rating WHERE title = 'The Hobbit';
```

You should get the following output:

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> SELECT
EXPLODE(context_ngrams(sentences(LOWER(r_review)),
. . . . . . . . . . . . . . . . . . . . . . .> array(null, null, null), 10)) AS
trigram FROM books_rating WHERE title = 'The Hobbit';
```

```
----------------------------------------------------------------------------------------
--------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED
KILLED
----------------------------------------------------------------------------------------
--------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0
0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0
0
----------------------------------------------------------------------------------------
--------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 16.60 s
----------------------------------------------------------------------------------------

--------


+---------------------------------------------------+
|                      trigram                      |
+---------------------------------------------------+
| {"ngram":["lord","of","the"],"estfrequency":2670.0} |
| {"ngram":["of","the","rings"],"estfrequency":2457.0} |
| {"ngram":["the","lord","of"],"estfrequency":1871.0} |
| {"ngram":["the","hobbit","is"],"estfrequency":1693.0} |
| {"ngram":["read","this","book"],"estfrequency":1257.0} |
| {"ngram":["this","book","is"],"estfrequency":1020.0} |
| {"ngram":["read","the","hobbit"],"estfrequency":939.0} |
| {"ngram":["one","of","the"],"estfrequency":795.0}   |
| {"ngram":["hobbit","is","a"],"estfrequency":734.0}  |
| {"ngram":["this","is","a"],"estfrequency":685.0}    |
+---------------------------------------------------+
10 rows selected (22.103 seconds)
```

From this we can see a lot of tri-grams, to get further more context we now use context n-gram to find next 2 words after 'this' 'book' 'is' to get more insights by using the following code:

```
SELECT EXPLODE(context_ngrams(sentences(LOWER(r_review)),

array("this", "book", "is", null, null), 10)) AS fivegram FROM books_rating WHERE title = 'The Hobbit';
```

You should get the following output:

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> SELECT
EXPLODE(context_ngrams(sentences(LOWER(r_review)),
. . . . . . . . . . . . . . . . . . . . . . .> array("this", "book", "is", null,
null), 10)) AS fivegram FROM books_rating WHERE title = 'The Hobbit';
----------------------------------------------------------------------------------------
--------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED
KILLED
----------------------------------------------------------------------------------------
--------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0
0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0
0
----------------------------------------------------------------------------------------
--------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 18.45 s
----------------------------------------------------------------------------------------
--------
```

```
+-----------------------------------------------+
|                    fivegram                   |
+-----------------------------------------------+
| {"ngram":["about","a"],"estfrequency":50.0}    |
| {"ngram":["one","of"],"estfrequency":35.0}     |
| {"ngram":["a","must"],"estfrequency":24.0}     |
| {"ngram":["a","classic"],"estfrequency":20.0}  |
| {"ngram":["a","great"],"estfrequency":20.0}    |
| {"ngram":["the","best"],"estfrequency":20.0}   |
| {"ngram":["a","true"],"estfrequency":19.0}     |
| {"ngram":["a","wonderful"],"estfrequency":16.0} |
| {"ngram":["amazing","i"],"estfrequency":16.0}  |
| {"ngram":["a","timeless"],"estfrequency":15.0} |
+-----------------------------------------------+
10 rows selected (19.078 seconds)
```

From this you can continue futher with sixth gram and so on, with this fivegram we can say that most of them say it's a classic, great, best, wonderful etc… which is all positive.

Now for least10 rated books i.e 'Red Rabbit'

```
SELECT EXPLODE(NGRAMS(SENTENCES(LOWER(r_summary)), 2, 5))

AS bigrams

FROM books_rating

WHERE title = 'Red Rabbit';
```

You get the following output:

```
+----------------------------------------------+
|                    bigrams                   |
+----------------------------------------------+
| {"ngram":["red","rabbit"],"estfrequency":54.0} |
| {"ngram":["tom","clancy"],"estfrequency":28.0} |
| {"ngram":["jack","ryan"],"estfrequency":21.0}  |
| {"ngram":["of","the"],"estfrequency":11.0}     |
| {"ngram":["the","worst"],"estfrequency":9.0}   |
+----------------------------------------------+
5 rows selected (21.182 seconds)
```

From this we for futher analysis this time lets see the summary and review where we have the words 'the worst' in it, run the following code:

```
SELECT r_summary, r_review

FROM books_rating

WHERE title = 'Red Rabbit'

AND r_summary LIKE '%the worst%' LIMIT 3;
```

You get the following output:

```
+----------------------------------------------------+----------------------------------
--------------------+
|                     r_summary                      |                        r_review
|
+----------------------------------------------------+----------------------------------
--------------------+
| By far the worst Jack Ryan book                    | "I thought long and hard before
writing this review. I knew my complaints with Red Rabbit had much to do with my
knowledge of what Clancy was capable of when working at full capacity |
| One of the worst books I've ever struggled through. | "I like Tom Clancy. With the
exception of the new hardcover now out |
| Horrible.... the worst from Clancy EVER            | "This book is horrible... even
the Bear and the Dragon was better than this (and I also disliked that one). Even if
we ignore Clancy's CONSTANT superficial right-wing rhetoric (not that is a bad
political stance |
+----------------------------------------------------+----------------------------------
--------------------+
3 rows selected (16.806 seconds)
```

We see that the same author has written another book ' the Bear and the Dragon' which is in fact 3rd least rated book from our analysis. This review backup out findings that 'red rabbit' is least rated.

4. Sentiment Analysis for book_rating by using dictionary to get the sentiments of review text and review summary columns. Use the following 3 Hive commands to create 3 views that will allow us to do that:

```
-- Create view l1 to compute sentiment for review text column
create view IF NOT EXISTS l1 as
        select id, user_id, words
        from books_rating
        lateral view explode(sentences(lower(r_review))) dummy as words;

-- Create view l2 from l1 to compute sentiment
create view IF NOT EXISTS l2 as
        select id, user_id, word
        from l1
        lateral view explode(words) dummy as word;

-- Create view l3 from l2 to compute sentiment
create view IF NOT EXISTS l3 as select
        id,
        user_id,
        l2.word,
        case d.polarity
        when 'negative' then -1
        when 'positive' then 1
        else 0 end as polarity
        from l2 left outer join dictionary d on l2.word = d.word;
```

Viewing the contents of l3 by using the following code:

```
select * from l3 limit 3;
```

You get the following output:

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> select * from l3 limit 3;
+-------------+---------------+----------+--------------+
|    l3.id    |   l3.user_id  | l3.word  | l3.polarity  |
+-------------+---------------+----------+-------------+
| 1882931173  | AVCGYZL8FQQTD  | this     | 0           |
| 1882931173  | AVCGYZL8FQQTD  | is       | 0           |
| 1882931173  | AVCGYZL8FQQTD  | only     | 0           |
+-------------+---------------+----------+-------------+
3 rows selected (10.451 seconds)
```

We create an external table called review_sentiment to sum all sentiments by the review id to get the sentiment of that review by using the below code:

```
create external table IF NOT EXISTS review_sentiment(id bigint,user_id bigint, r_sentiment int)
stored as orc;

INSERT OVERWRITE TABLE review_sentiment select

 id,user_id,

 case

 when sum( polarity ) > 0 then '2'

 when sum( polarity ) < 0 then '0'

 else '1' end as sentiment

from l3 group by id,user_id;


select * from review_sentiment limit 3;
```

After successfully running these codes, when you run the select statement, you'll get the following output:

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> select * from review_sentiment limit 3;
+----------------------+--------------------------+------------------------------+
| review_sentiment.id  | review_sentiment.user_id  | review_sentiment.r_sentiment  |
+----------------------+--------------------------+------------------------------+
| NULL                 | NULL                     | 1                            |
| NULL                 | NULL                     | 1                            |
| NULL                 | NULL                     | 1                            |
+----------------------+--------------------------+------------------------------+
3 rows selected (0.495 seconds)
```

Do the same for summery text column by running the following codes:

```
-- Create view t1 to compute sentiment for summary text column
create view IF NOT EXISTS t1 as
 select id, user_id, words
 from books_rating
 lateral view explode(sentences(lower(r_summary))) dummy as words;

-- Create view t2 from t1 to compute sentiment
create view IF NOT EXISTS t2 as
 select id, user_id, word
 from t1
 lateral view explode(words) dummy as word;

-- Create view t3 from t2 to compute sentiment
create view IF NOT EXISTS t3 as select
 id,
 user_id,
 t2.word,
 case d.polarity
 when 'negative' then -1
 when 'positive' then 1
 else 0 end as polarity
 from t2 left outer join dictionary d on t2.word = d.word;
```

Viewing the contents of t3 by using the following code:

```
select * from t3 limit 3;
```

You get the following output:

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> select * from t3 limit 3;
+-------------+-----------------+-------------+--------------+
|    t3.id    |   t3.user_id    |   t3.word   | t3.polarity  |
+-------------+-----------------+-------------+--------------+
| 1882931173  | AVCGYZL8FQQTD   | nice        | 1            |
| 1882931173  | AVCGYZL8FQQTD   | collection  | 0            |
| 1882931173  | AVCGYZL8FQQTD   | of          | 0            |
+-------------+-----------------+-------------+--------------+
3 rows selected (4.824 seconds)
```

We create an external table called summary_sentiment to sum all sentiments by the review id to get the sentiment of that review by using the below code:

```
create external table IF NOT EXISTS summary_sentiment(id bigint, user_id bigint, s_sentiment int)
stored as orc;

INSERT OVERWRITE TABLE summary_sentiment select

 id, user_id,

 case

 when sum( polarity ) > 0 then '2'

 when sum( polarity ) < 0 then '0'

 else '1' end as sentiment

from t3 group by id, user_id;


select * from summary_sentiment limit 3;
```

After successfully running these codes, when you run the select statement, you'll get the following output:

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai>select * from summary_sentiment limit 3;
+-----------------------+---------------------------+--------------------------------
+
| summary_sentiment.id  | summary_sentiment.user_id  | summary_sentiment.s_sentiment
|
+-----------------------+---------------------------+--------------------------------
+
| NULL                  | NULL                      | 1
|
| NULL                  | NULL                      | 1
|
| NULL                  | NULL                      | 1
|
+-----------------------+---------------------------+--------------------------------
+
3 rows selected (0.518 seconds)
```

Joining both review sentiment and summary sentiment by leftjoin over id and user_is using following code:

```
create view IF NOT EXISTS rs_sentiments as select

 r.id, r.user_id, r.r_sentiment, s.s_sentiment from review_sentiment r left outer join
 summary_sentiment s on r.id = s.id and r.user_id = s.user_id;



select * from rs_sentiments limit 3;
```

Let's see the output:

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> select * from rs_sentiments limit 3;
```

```
+-------------------+-------------------------+---------------------------+----------
-----------------+
| rs_sentiments.id  | rs_sentiments.user_id  | rs_sentiments.r_sentiment  |
rs_sentiments.s_sentiment  |
+-------------------+-------------------------+---------------------------+----------
-----------------+
| NULL              | NULL                    | 1                          | NULL
|
| NULL              | NULL                    | 1                          | NULL
|
| NULL              | NULL                    | 1                          | NULL
|
+-------------------+-------------------------+---------------------------+----------
-----------------+
3 rows selected (8.426 seconds)
```

Joining the review, summary sentiment and, books_rating into another table by creating an external book_senti table using the below code:

```
create external table IF NOT EXISTS books_senti(id bigint, title string, r_date timestamp, r_score
float, r_sentiment int, s_sentiment int) stored as orc;


INSERT OVERWRITE TABLE books_senti select
 s.id,
 r.title,
 cast(from_unixtime(r_time) as timestamp) r_date,
 r.r_score,
 s.r_sentiment,
 s.s_sentiment
FROM books_rating r LEFT OUTER JOIN rs_sentiments s on r.id = s.id;


Select * from books_senti limit 1;
```

The output will be:

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> Select * from books_senti limit 1;
+-----------------+---------------------------------+-----------------------+--------
--------------+-------------------------+-------------------------+
| books_senti.id  |        books_senti.title        |   books_senti.r_date   |
books_senti.r_score  | books_senti.r_sentiment  | books_senti.s_sentiment  |
+-----------------+---------------------------------+-----------------------+--------
--------------+-------------------------+-------------------------+
| 1882931173      | Its Only Art If Its Well Hung!  | 1999-10-23 00:00:00.0  | 4.0
| 2                        | NULL                     |
+-----------------+---------------------------------+-----------------------+--------
--------------+-------------------------+-------------------------+
1 row selected (0.388 seconds)
```

Run these two queries to see the deviation in sentiment tone for same book review and its summary:

```
select count(*) from books_senti;

select count(*) from books_senti where review_sentiment != summary_sentiment;
```

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> select count(*) from books_senti;
+----------+
|   _c0    |
+----------+
| 3252448  |
+----------+
1 row selected (26.628 seconds)
```

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> select count(*) from books_senti where
review_sentiment != summary_sentiment;

+---------+
|   _c0   |
+---------+
| 298047  |
+---------+
1 row selected (22.341 seconds)
```

From this you can see that in 3.2M reviews about 300k reviews have different sentiment for the same books review text and it’s review summary column, this shows the limitations of our sentiment analysis where in it is difficult for machine to get 100% accurate sentiment. Major problem for this being double negation and sarcastic reviews which are sometimes even difficult for humans to understand.

5. In order to clean the data by eliminating non-alphabetical and non-numeric characters for visualization, we create one final table books_review and a view for top25 books, we join the table to only get the sentiments for all these top25books by using the below code:

```
create view top25 as
select * from top10books order by count DESC limit 25;

CREATE EXTERNAL TABLE IF NOT EXISTS books_review ( id bigint, title string, r_date timestamp,
r_score float, r_sentiment int, s_sentiment int)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ","
STORED AS TEXTFILE
LOCATION "/user/ychauha4/tmp/books_reviews";


INSERT OVERWRITE TABLE books_review
 select b.id,
 REGEXP_REPLACE(b.title, '[^a-zA-Z0-9]+', ' ') title,
 b.r_date, b.r_score,
 b.review_sentiment, b.summary_sentiment from books_senti b LEFT SEMI JOIN top25 t on b.title =
t.title
 where id IS NOT NULL
 AND ( review_sentiment =0 or review_sentiment =1 or review_sentiment =2)
 AND ( summary_sentiment =0 or summary_sentiment =1 or summary_sentiment =2);
```

You'll get the following output:

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> create view top25 as
. . . . . . . . . . . . . . . . . . . . . . .> select * from top10books order by count
DESC limit 25;
No rows affected (0.679 seconds)


0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> INSERT OVERWRITE TABLE books_review
. . . . . . . . . . . . . . . . . . . . . . .> select b.id,
. . . . . . . . . . . . . . . . . . . . . . .> REGEXP_REPLACE(b.title, '[^a-zA-Z0-
9]+', ' ') title,
. . . . . . . . . . . . . . . . . . . . . . .> b.r_date, b.r_score,
. . . . . . . . . . . . . . . . . . . . . . .> b.review_sentiment, b.summary_sentiment
from books_senti b LEFT SEMI JOIN top25 t on b.title = t.title
. . . . . . . . . . . . . . . . . . . . . . .> where id IS NOT NULL
. . . . . . . . . . . . . . . . . . . . . . .> AND ( review_sentiment =0 or
review_sentiment =1 or review_sentiment =2)
. . . . . . . . . . . . . . . . . . . . . . .> AND ( summary_sentiment =0 or
summary_sentiment =1 or summary_sentiment =2);
--------------------------------------------------------------------------------------
--------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED
KILLED
--------------------------------------------------------------------------------------
--------
Map 3 .......... container     SUCCEEDED      1          1        0        0       0
0
```

```
Map 4 .......... container     SUCCEEDED      1          1         0         0        0
0
Map 5 .......... container     SUCCEEDED      1          1         0         0        0
0
Reducer 6 ...... container     SUCCEEDED     21         21         0         0        0
0
Reducer 7 ...... container     SUCCEEDED      1          1         0         0        0
0
Map 1 .......... container     SUCCEEDED      1          1         0         0        0
0
Reducer 2 ...... container     SUCCEEDED      1          1         0         0        0
0
--------------------------------------------------------------------------------------
--------
VERTICES: 07/07  [==========================>>] 100%  ELAPSED TIME: 61.71 s
--------------------------------------------------------------------------------------
--------
No rows affected (68.8 seconds)

0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> select * from books_review limit 3;
+-------------------+---------------------+-------------------------+-------------------
----+--------------------------+-------------------------+
| books_review.id  | books_review.title  |  books_review.r_date    |
books_review.r_score  | books_review.r_sentiment  | books_review.s_sentiment  |
+-------------------+---------------------+-------------------------+-------------------
----+--------------------------+-------------------------+
| 736641238        | Little Women        | 2004-02-05 00:00:00.0  | 5.0
| 2                         | 2                      |
| 736641238        | Little Women        | 2004-01-14 00:00:00.0  | 4.0
| 2                         | 2                      |
| 736641238        | Little Women        | 2003-12-10 00:00:00.0  | 4.0
| 2                         | 2                      |
+-------------------+---------------------+-------------------------+-------------------
----+--------------------------+-------------------------+
3 rows selected (0.419 seconds)
```

Step 3: Downloading and Visualization

Now that we have done all our analysis and quering in Hive/beeline it's now time to download the files from HDFS to out linux to out local PC by using -get and SCP commands as foolows:

-get commands

```
$ hdfs dfs -get tmp/books_reviews/000000_0  books_reviews.csv

$ hdfs dfs -get tmp/top10/000000_0 top10reviewed.csv

$ hdfs dfs -get tmp/least10/000000_0 least10rated.csv

$ hdfs dfs -get tmp/top10categories/000000_0 top10categories.csv
```

Check in Linux system with ls command as follows:

```
-bash-4.2$ ls
books_reviews.csv  least10rated.csv  top10categories.csv
dictionary.tsv     __MACOSX          top10reviewed.csv
```

As you can see the files have been downloaded to Linux server, now let's open another GitBash terminal, don't ssh and using SCP we download this to out personal computer for visualization:

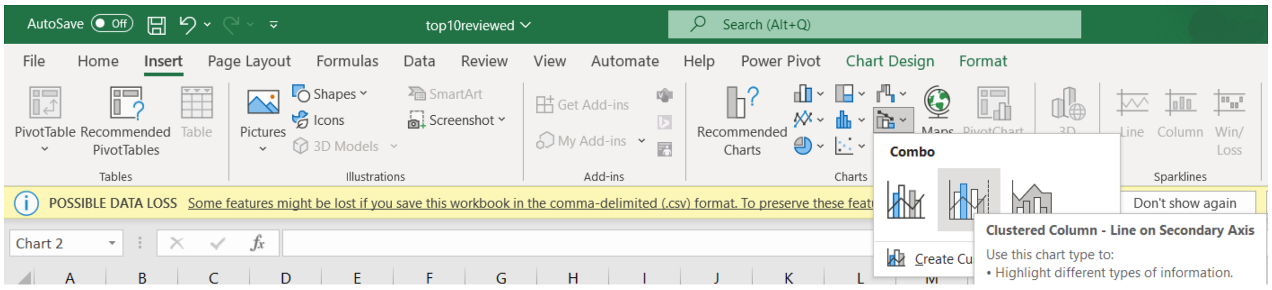
```
$ scp ychauha4@144.24.14.145:/home/ychauha4/books_reviews.csv .

$ scp ychauha4@144.24.14.145:/home/ychauha4/top10reviewed.csv .

$ scp ychauha4@144.24.14.145:/home/ychauha4/least10rated.csv .

$ scp ychauha4@144.24.14.145:/home/ychauha4/top10categories.csv .
```

Note – Replace ychauha4 with your username and IP address with the current IP. You will be asked password for every download.

```
$ scp ychauha4@144.24.14.145:/home/ychauha4/books_reviews.csv .
ychauha4@144.24.14.145's password:
books_reviews.csv                             100% 2095KB   1.8MB/s   00:01

$ scp ychauha4@144.24.14.145:/home/ychauha4/top10reviewed.csv .
ychauha4@144.24.14.145's password:
top10reviewed.csv                             100%  416     5.2KB/s   00:00

$  scp ychauha4@144.24.14.145:/home/ychauha4/least10rated.csv .
ychauha4@144.24.14.145's password:
least10rated.csv                              100%  470     3.8KB/s   00:00

$ scp ychauha4@144.24.14.145:/home/ychauha4/top10categories.csv .
ychauha4@144.24.14.145's password:
top10categories.csv                           100%  232     5.2KB/s   00:00
```

Now to Visualize all these 4 documents in Excel and Tableau.

1. First let's start with out top10reviewed.csv file, open your Excel first then in Excel click on open file, navigate to your PC's file stored location in my case it is 'C:\Users\Yougender Chauhan'. Open the top10reviewed.csv.


AutoSave Off
top10reviewed
Search (Alt+Q)
File Home Insert Page Layout Formulas Data Review View Automate Help Power Pivot Chart Design Format
PivotTable Recommended PivotTables Table
Tables
Pictures Shapes Icons 3D Models SmartArt Screenshot
Illustrations
Get Add-ins My Add-ins
Add-ins
Recommended Charts
Charts
Combo
Create Cu
Line Column Win/Loss
Sparklines
Chart 2
A B C D E F G H I J K L M

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these feat

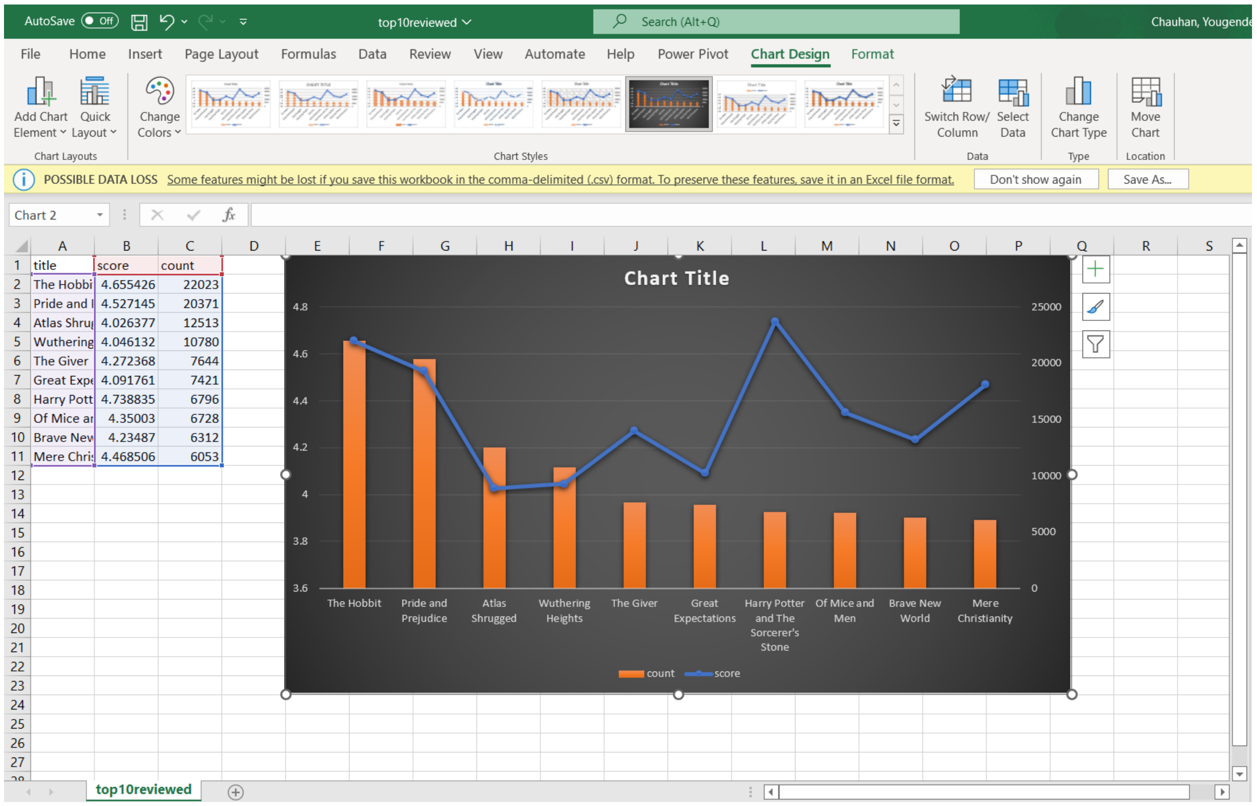
Don't show again

Clustered Column - Line on Secondary Axis

Use this chart type to:

• Highlight different types of information.

- After opening the excel, insert one top row, and add column names: title, score and count. Select all the filled rows and column, on top bar, click on 'insert' then on the right side, select Combo and then clustered column chart.



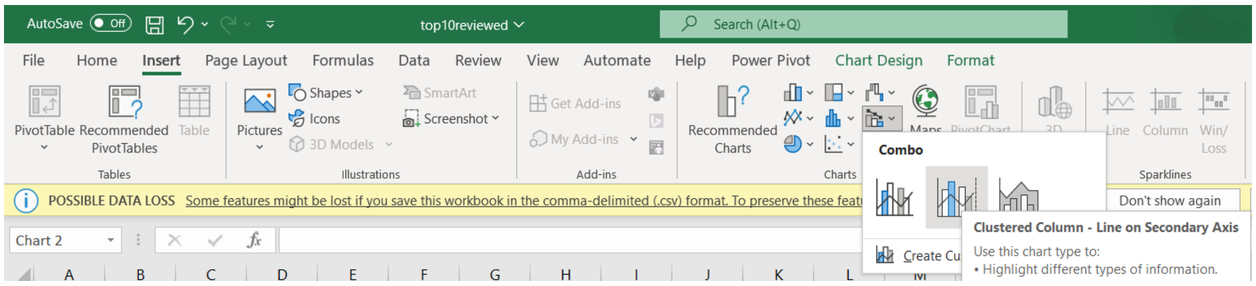
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

Don't show again

	A	B	C
1	title	score	count
2	The Hobbi	4.655426	22023
3	Pride and I	4.527145	20371
4	Atlas Shruş	4.026377	12513
5	Wuthering	4.046132	10780
6	The Giver	4.272368	7644
7	Great Expe	4.091761	7421
8	Harry Pott	4.738835	6796
9	Of Mice ar	4.35003	6728
10	Brave Nev	4.23487	6312
11	Mere Chri	4.468506	6053

- This chart shows the Number of reviews for these 10 books and the line their respective average score.

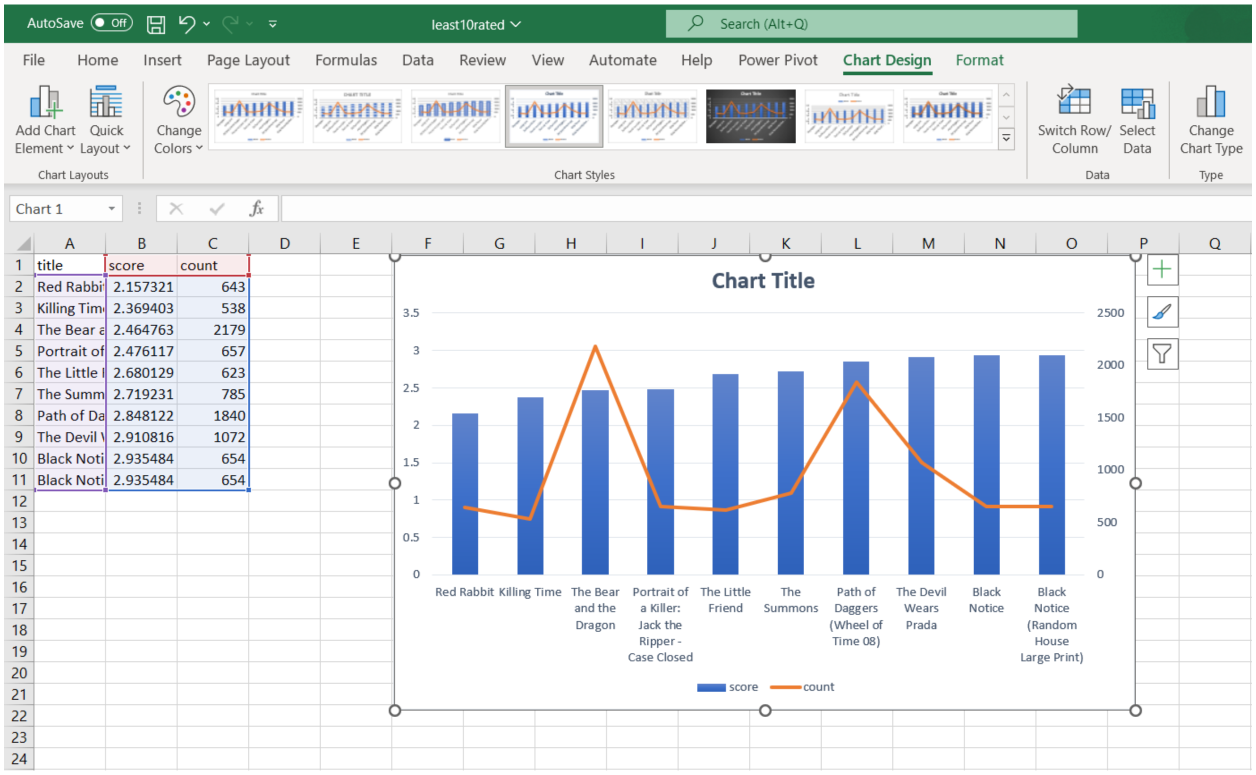
2. Next, lets do the same for least10 rated books by following the above same steps. first then in Excel click on open file, navigate to your PC's file stored location in my case it is 'C:\Users\Yougender Chauhan'. Open the least10rated.csv.



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these feat

Clustered Column - Line on Secondary Axis
Use this chart type to:
• Highlight different types of information.

- After opening the excel, insert one top row, and add column names: title, score and count. Select all the filled rows and column, on top bar, click on 'insert' then on the right side, select Combo and then clustered column chart.

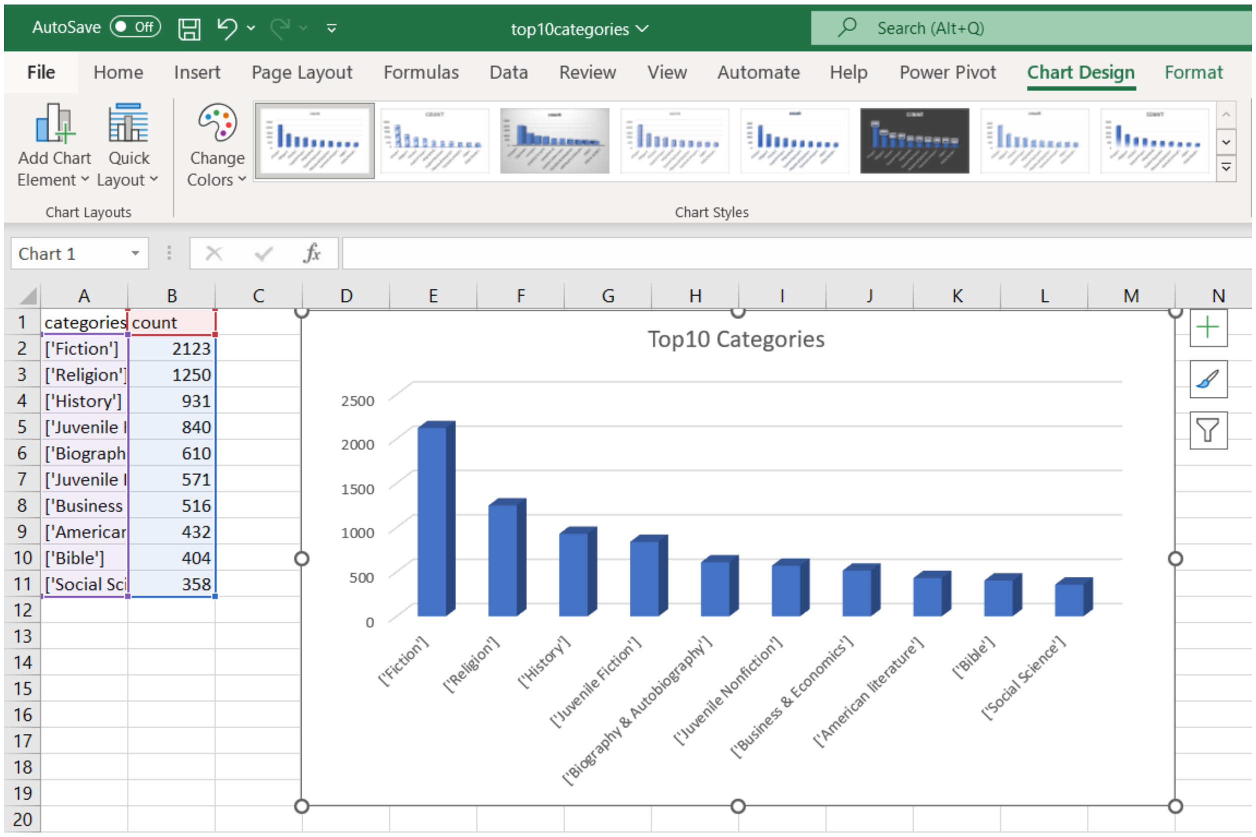


title	score	count
Red Rabbi	2.157321	643
Killing Tim	2.369403	538
The Bear a	2.464763	2179
Portrait of	2.476117	657
The Little I	2.680129	623
The Summ	2.719231	785
Path of Da	2.848122	1840
The Devil \	2.910816	1072
Black Noti	2.935484	654
Black Noti	2.935484	654

- This chart shows us the score and the line represent its respective number of reviews for that book.

3. Now for the top 10 categories of unique books present in our dataset. Open excel, click on open file then navigate to your PC's file stored location in my case it is 'C:\Users\Yougender Chauhan'. Open the top10categories.csv.

- After opening the excel, insert one top row, and add column names: categories and count. Select all the filled rows and column, on top bar, click on 'insert' then on the right side, select Bar Chart.

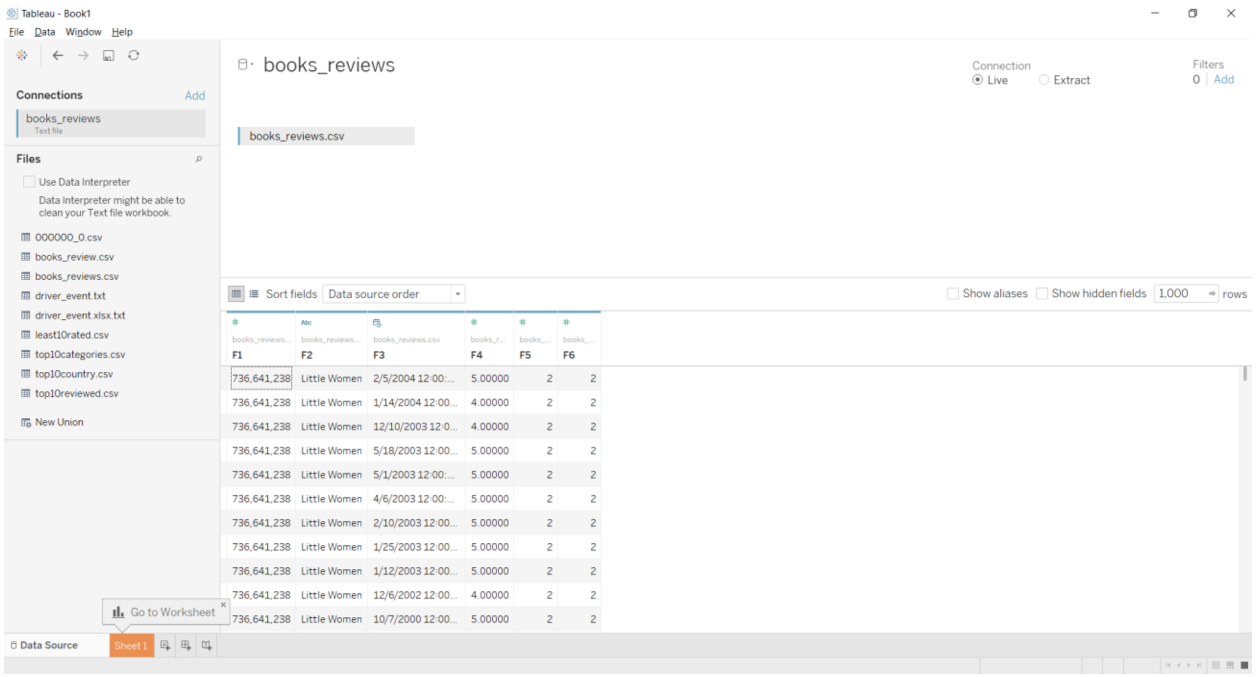


	A	B
1	categories	count
2	['Fiction']	2123
3	['Religion']	1250
4	['History']	931
5	['Juvenile I	840
6	['Biograph	610
7	['Juvenile I	571
8	['Business	516
9	['Americar	432
10	['Bible']	404
11	['Social Sci	358

- This chart shows us the count of categories of each unique book.

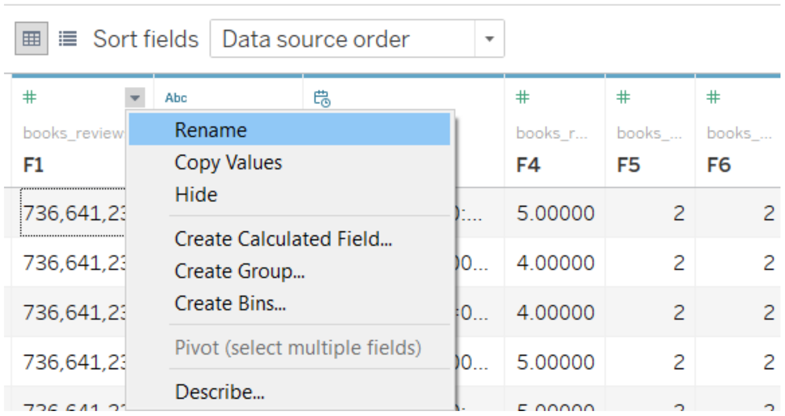
4. Finally let's use tableau to visualize out sentiments that we have downloaded as 'books_reviews.csv'. to do so, first open then click on text file from there go to the path where the file is saved, in my case it is at 'C:\Users\Yougender Chauhan'. Open 'books_reviews.csv'.

- When you load the data, it looks like this:



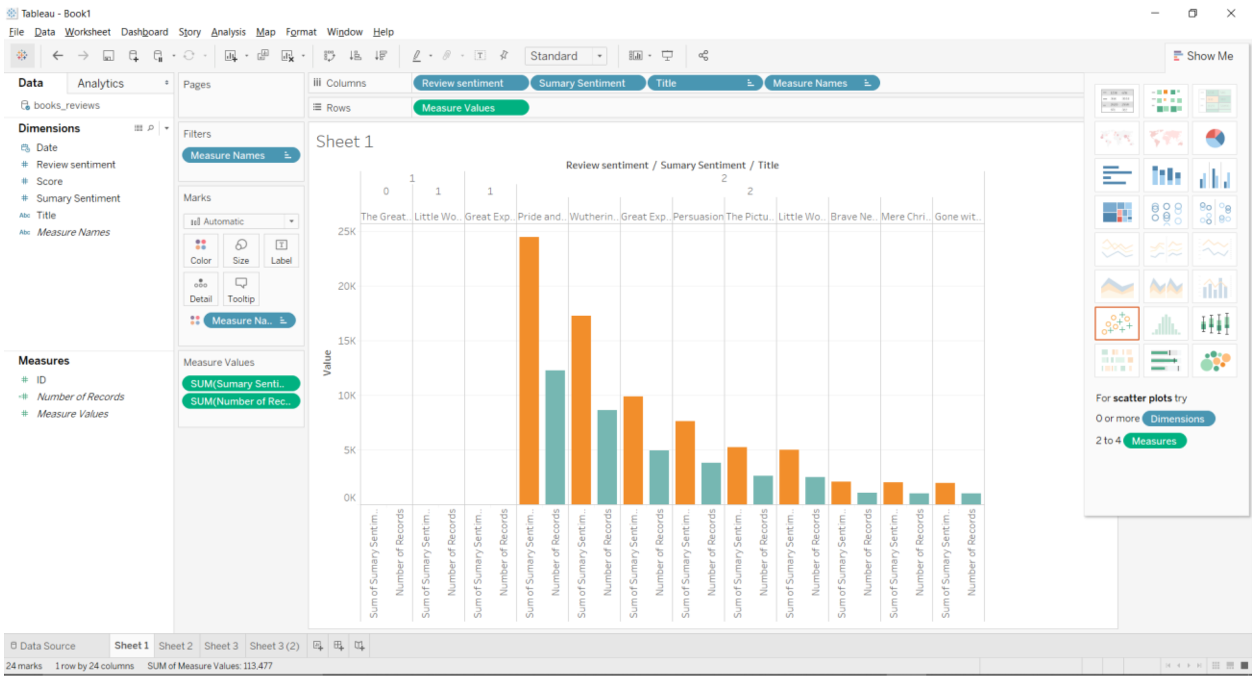
books_reviews... F1	books_reviews.... F2	books_reviews.csv F3	books_r... F4	books_... F5	books_... F6
736,641,238	Little Women	2/5/2004 12:00:...	5.00000	2	2
736,641,238	Little Women	1/14/2004 12:00...	4.00000	2	2
736,641,238	Little Women	12/10/2003 12:0...	4.00000	2	2
736,641,238	Little Women	5/18/2003 12:00...	5.00000	2	2
736,641,238	Little Women	5/1/2003 12:00:...	5.00000	2	2
736,641,238	Little Women	4/6/2003 12:00:...	5.00000	2	2
736,641,238	Little Women	2/10/2003 12:00...	5.00000	2	2
736,641,238	Little Women	1/25/2003 12:00...	5.00000	2	2
736,641,238	Little Women	1/12/2003 12:00...	5.00000	2	2
736,641,238	Little Women	12/6/2002 12:00...	4.00000	2	2
736,641,238	Little Women	10/7/2000 12:00...	5.00000	2	2

- Here click on dropdown on the column name F1 and click 'rename' and rename F1 to 'ID', F2 to 'Title', F3 to 'Date', F4 to 'Score', F5 to 'Review sentiment', F6 to 'Summary sentiment'. After which click on Sheet 1.

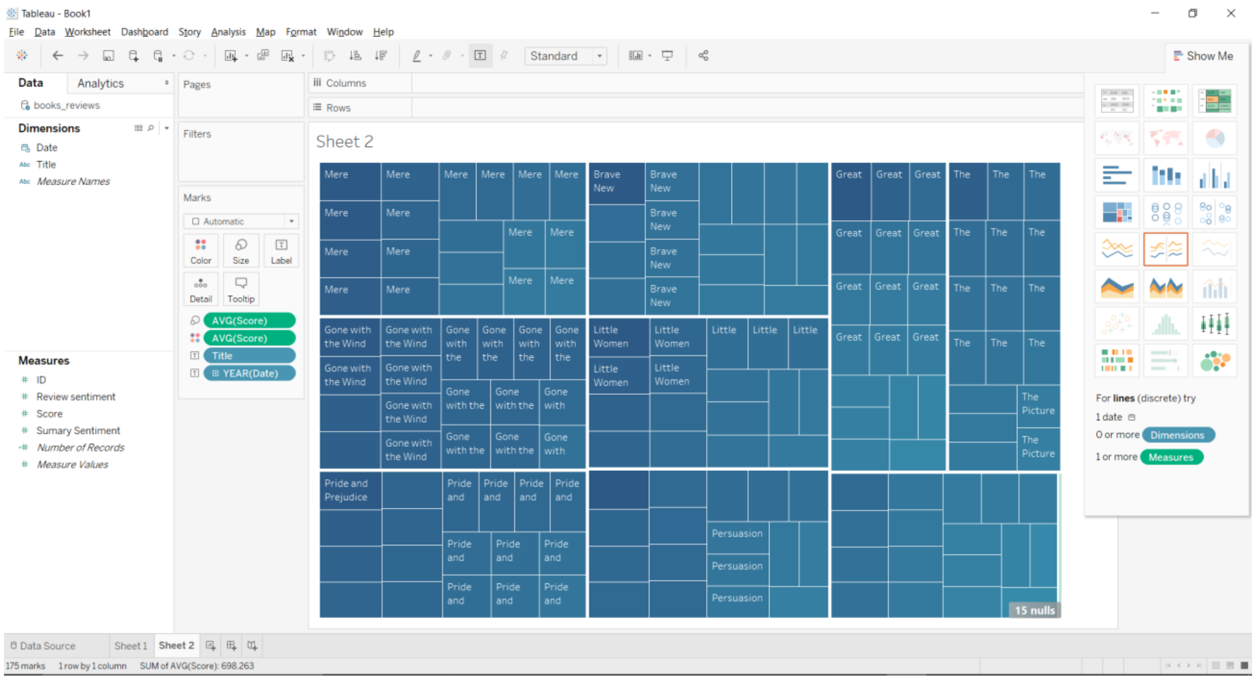


books_review F1	F4	F5	F6
736,641,23	5.00000	2	2
736,641,23	4.00000	2	2
736,641,23	4.00000	2	2
736,641,23	5.00000	2	2

- Drag and drop 'Title' from Dimensions to Columns and 'Review Sentiment', 'Summary Sentiment' to Rows field. Move 'review sentiment' and 'summary sentiment' to dimensions and add to columns, then from Show Me drop down on right side, select side-by-side bar chart click on sort button from the top toolbar you get the following:

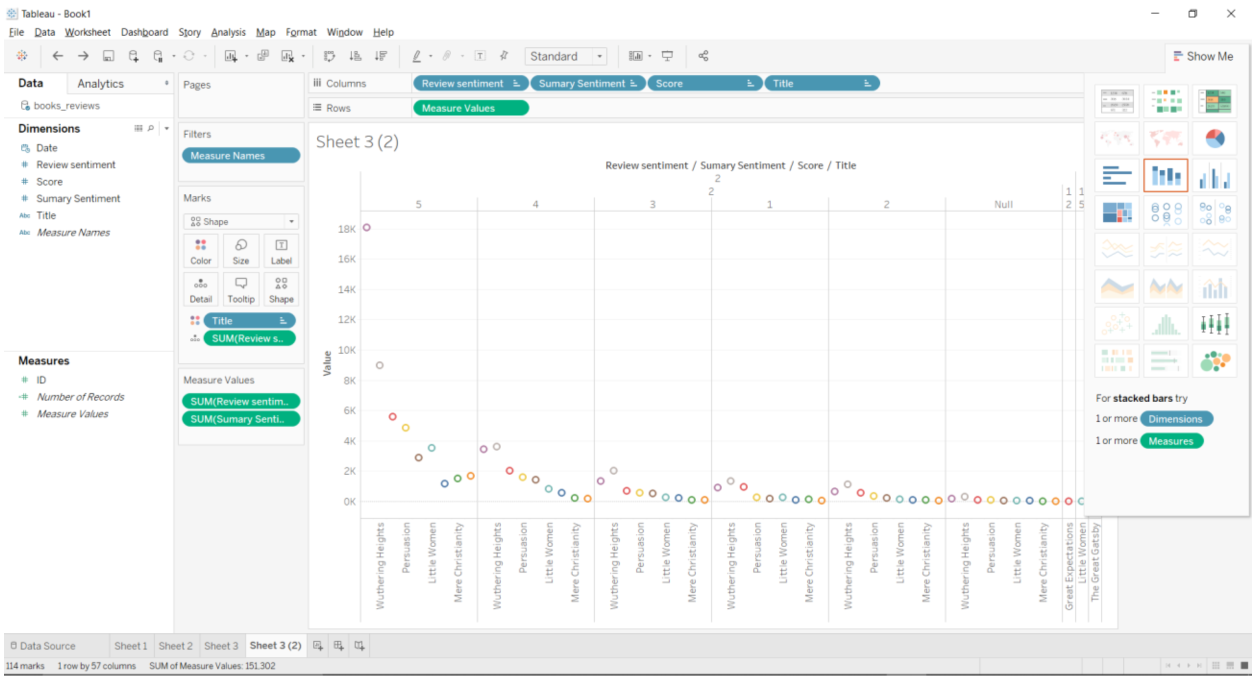


- Now, click on new worksheet beside 'sheet 1' on the bottom. On column add 'Title' and 'Date' on Rows add 'Score' then select treemaps from the show me dropdown on the right. Then in 'Marks' panel, sum(score) > measure(sum) > average. Do the same for color as well.



- You will see this treemaps which shows us the title and their average review score based on the year of review written.

- Next, click on new worksheet beside 'sheet 2' on the bottom. Move 'review sentiment' and 'summary sentiment' to dimensions, add 'score', 'title', 'review sentiment', 'summary sentiment'. Move 'review sentiment' and 'summary sentiment' to Measure and add it to rows. Select side-by-side circles on Show Me dropdown and sort by Title.



- This shows us the summary and review sentiments and the avg score side by side to see any deviations. Since we took a very small sample set for this example, we don't see the vast deviation here.

Summary

In this Tutorial you learned how to use Hadoop and Hive to do Sentiment analysis, Data analysis. We were also able to cover the limitation of using our current sentiment analysis model and using Excel and Tableau we visualized out analyzed data for better understanding.

References

1. URL of Data Source, https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews?select=Books_rating.csv

2. URL of GitHub, https://github.com/Chauhan67/Amazonbookreviews

3. Lab tutorials used:

    - labSentimentAnalysisTextNgrams
    - labTwitterSentimentAnalysisLab
    - labTableau_oracle_v2
    - lab2HiveSensorDataAnalysisLab_aws