

# Lab Tutorial

GROUP 2, Section 2

Sahavat Tangchitnob ([stangch2@calstatela.edu](mailto:stangch2@calstatela.edu))

Maria Espinoza ([mespino6@calstatela.edu](mailto:mespino6@calstatela.edu))

Julio Rojas ([jrojas2@calstatela.edu](mailto:jrojas2@calstatela.edu))

Swarnim Jambhule ([sjambhu@calstatela.edu](mailto:sjambhu@calstatela.edu))

12/12/2018

## Amazon Data Analysis using HIVE

### Objectives

1. Download, unzip and upload data to HDFS
2. Create Hive tables to query Amazon data
3. Create Hive queries to analyze the data
4. Create Hive queries to analyze sentiment of data using dictionary
5. Download Data into your PC
6. Analyze Data using Power BI

### Platform Spec

Oracle Big Data Compute Edition: 5 nodes

CPU Speed: 2.2 GHz

OCPUs: 10

Memory: 150 GB

Storage: 678 GB

HDFS Capacity: 147 GB

### STEP 1: Remotely connect to Oracle Cloud

You must have an ip address to connect to Oracle Cloud.

Your CalStateLA username should be a username / password for Oracle account.

1. SSH to connect it, for example, the user mespino6:

ssh mespino6@129.150.128.177

```
Sahavats-Air:~ sahavattangchitnob$ ssh mespino6@129.150.128.177
mespino6@129.150.128.177's password:
-bash-4.1$
```

You may run the following HDFS commands to test if hdfs works well on your Oracle account:

```
hdfs dfs -ls
hdfs dfs -mkdir test
hdfs dfs -ls
```

Example:

```
-bash-4.1$ hdfs dfs -mkdir test;
-bash-4.1$ hdfs dfs -ls;
Found 7 items
drwxr-xrwx - mespino6 hdfs 0 2018-10-03 01:46 .hiveJars
drwx----- - mespino6 hdfs 0 2018-11-28 03:50 .staging
drwxr-xrwx - mespino6 hdfs 0 2018-09-26 03:12 SensorFiles
drwxr-xr-x - mespino6 hdfs 0 2018-11-28 02:33 dualcore
drwxr-xr-x - mespino6 hdfs 0 2018-11-08 22:25 output
drwxr-xr-x - mespino6 hdfs 0 2018-12-03 04:05 test
drwxr-xrwx - mespino6 hdfs 0 2018-12-01 01:22 tmp
```

2. Now you have the following 3 commands. The first is to create a directory named "data". The second is to create a directory named "tables" inside tmp/data/.The third is to list the the files and folders of /user/mespino6/tmp/data/tables.

```
-bash-4.1$ hdfs dfs -mkdir tmp/data
-bash-4.1$ hdfs dfs -mkdir tmp/data/tables
-bash-4.1$ hdfs dfs -ls tmp/data/tables
```

3. Run the following HDFS command to make your beeline command works:

```
-bash-4.1$ hdfs dfs -chmod -R o+w tmp/
```

Example:

```
-bash-4.1$ hdfs dfs -chmod -R o+w tmp/
-bash-4.1$ hdfs dfs -ls tmp/
Found 1 items
drwxr-xrwx _ - mespino6 hdfs 0 2018-12-01 01:22 tmp/data
```

## **STEP 2: Downloading Data into your Oracle Big Data**

After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows:

1. Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file and unzip the contents to - at the HDFS path “/user/mespino6/tmp/data/tables ”:

```
wget -O amazon_reviews_us_Books_v1_02.tsv.gz  
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Books_v1_02  
.tsv.gz
```

2. Unzip and put tsv.gz into hdfs:

```
gunzip -c amazon_reviews_us_Books_v1_02.tsv.gz | hadoop fs -put -  
/user/mespino6/tmp/data/tables;
```

## **STEP 3: Creating Hive Tables and Queries to Analyze Data**

1. Open beeline CLI (Command Line Shell Interface) that is equivalent to hive CLI environment as follows, which you have done in the previous lab. Beeline is for multiple users' access to Hive Server 2 of a Hadoop cluster. You have to copy and paste “!connect ...” command given by the instructor at the lab page of Canvas to beeline and press enter without any password when it asks for password.

```
-bash-4.1$ beeline
```

NOTE: the following connect url is an example and it should be given by the instructor at a lab page of the course web site:

WARNING: Use "yarn jar" to launch YARN applications.

Beeline version 1.2.1000.2.4.2.0-258 by Apache Hive

```
beeline> !connect
```

```
jdbc:hive2://cis5200-bdcsce-4.compute-608214094.oraclecloud.internal:2181,cis5200-b  
dcsce-2.compute-608214094.oraclecloud.internal:2181,cis5200-bdcsce-3.compute-608  
214094.oraclecloud.internal:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperName  
space=hiveserver2?tez.queue.name=interactive bdcsce_admin
```

It should read:

Connecting to

```
jdbc:hive2://cis5200-bdcsce-4.compute-608214094.oraclecloud.internal:2181,cis5200-bdcsce-2.compute-608214094.oraclecloud.internal:2181,cis5200-bdcsce-3.compute-608214094.oraclecloud.internal:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNameSpace=hiveserver2?tez.queue.name=interactive
```

Enter password for

```
jdbc:hive2://cis5200-bdcsce-4.compute-608214094.oraclecloud.internal:2181,cis5200-bdcsce-2.compute-608214094.oraclecloud.internal:2181,cis5200-bdcsce-3.compute-608214094.oraclecloud.internal:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNameSpace=hiveserver2?tez.queue.name=interactive:
```

Connected to: Apache Hive (version 1.2.1000.2.4.2.0-258)

Driver: Hive JDBC (version 1.2.1000.2.4.2.0-258)

Transaction isolation: TRANSACTION\_REPEATABLE\_READ

0: jdbc:hive2://cis5200-bdcsce-4.compute-6082>

NOTE: If you see "CLOSED" in the above beeline shell prompt, it is not connected to Hive Server2.

2. Now you have to create your database with your username to separate your tables with other users. For example, the user TtoTH should run the following:

Create database TtoTH;

use TtoTH;

```
0: jdbc:hive2://cis5200-bdcsce-4.compute-6082> use ttoth;  
No rows affected (0.168 seconds)
```

3. In the beeline shell CLI, you need to copy and paste the following HiveQL code to create an external table "amazon\_reviews\_traditional" and populate it with data you downloaded:

```
CREATE EXTERNAL TABLE amazon_reviews_traditional (  
  marketplace string,  
  customer_id string,  
  review_id int,
```

```

product_id int,
product_parent string,
product_title string,
product_category string,
star_rating int,
helpful_votes int,
total_votes int,
vine string,
verified_purchase string,
review_headline string,
review_body string,
review_date bigint,
year int)

```

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION  
'/user/mespino6/tmp/data/tables';

4. Now you can query the content of the amazon\_reviews\_traditional table:

```
SELECT * FROM amazon_reviews_traditional limit 10;
```

```

| marketplace | product_parent | NULL | verified_purchase | customer_id | product_title | NULL | review_headline | review_id | NULL | product_id | product_category | vine
| US | 122662979 | 4 | N | 12076615 | Sisterhood of the Traveling Pants (Book 1) | 2 | this book was a great learning novel! | 3 | 0385730586 | Books | N
you could learn from. it not only teaches the importance of family and their values but it also deals with basic issues that teens and some kids even deal
with. this book is about 4 best friends who are for the first time in their lives spending their summer apart. one day they are all in one of the girls roo
ms and finds a pair of pants that were tucked away in her closet. once all four of them try them on they realize that there is really something special ab
out these pants. seeming as how all 4 girls are differnt shapes and sizes and somehow the pants fit all of them, they realize that these pants were the st
art of something special. immediatley following they decided to make up certian rules about the pants such as you must write the best thing u did while wear
ing the pants over your summer on the right leg and also some silly things such as to \"never pick yuor nose while wearing the pants.\" this book follows
the girls throuh their summers in differnt places of the world and through all of the different obstacles that life takes them through. it can really teach
you alot not only about what is going on around you but most imporantly about yuorself. i would give this book 4 stars and would reccommend it to anyone
who seems the slighgtest bit interested.

| US | 56191234 | 3 | N | 12703090 | The Bad Girl's Guide to Getting What You Want | 5 | Fun Fluff | 5 | 0811828964 | Books | N
to stimulate your brain, this isn't it. However, if you are just looking for a good laugh, you'll enjoy The Bad Girl's Guide. It's funny and light, and d
efinitely a good way to pass a little bit of time.

```

Note: We created the `amazon_reviews_traditional` with columns using instructions from the location of the Amazon dataset:

To quickly get started with the dataset, in regions where [AWS Glue](#) is available you can use a nice feature called the [crawler](#) to automatically discover the data and create the required tables you will later query.

Alternatively you can head over to the [Amazon Athena console](#) and manually create a table as follows:

```
CREATE EXTERNAL TABLE amazon_reviews_parquet(  
  marketplace string,  
  customer_id string,  
  review_id string,  
  product_id string,  
  product_parent string,  
  product_title string,  
  star_rating int,  
  helpful_votes int,  
  total_votes int,  
  vine string,  
  verified_purchase string,  
  review_headline string,  
  review_body string,  
  review_date bigint,  
  year int)  
PARTITIONED BY (product_category string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  
LOCATION  
  's3://amazon-reviews-pds/parquet/'
```

Once the table is created execute the following in the Athena console only once:

As you can see with the results above, the `review_date` and `year` appears as `NULL`, thus, we cannot do tempo-spatial analysis. We assume that there is no year or review date data. Amazon dataset instructions:

5. Count the records in the ratings table to ensure that all 3105521 records are available:

```
SELECT COUNT(*) FROM amazon_reviews_traditional;
```

_c0
3105521

You can see the structure of the table as well:

```
DESCRIBE amazon_reviews_traditional;
```

col_name	data_type	comment
marketplace	string	
customer_id	string	
review_id	string	
product_id	string	
product_parent	string	
product_title	string	
product_category	string	
star_rating	int	
helpful_votes	int	
total_votes	int	
vine	string	
verified_purchase	string	
review_headline	string	
review_body	string	
review_date	bigint	
year	int	

5. We want to find the product that customers like most, but must guard against being misled by products that have few ratings assigned. Run the following query to find the product with the highest average using DESC among all those with at least 50 ratings, which should show the following result:

```
SELECT product_id, FORMAT_NUMBER(avg_star_rating,2) AS avg_star_rating
  FROM (SELECT product_id, AVG(star_rating) AS avg_star_rating,
              COUNT(*) AS num
        FROM amazon_reviews_traditional
        GROUP by product_id) amazon_reviews_traditional
 WHERE num >=50
 ORDER BY avg_star_rating DESC
 LIMIT 1;
```

product_id	avg_star_rating
0972217304	5.00

Rewrite, and then execute, the query above to find the product with the lowest average using ASC among products with at least 50 ratings (num >= 50). You should see that the result is product ID 007119551 with an average rating of 1.18, which should show the following result:



```

SELECT product_id, FORMAT_NUMBER(avg_star_rating,2) AS avg_star_rating
      FROM (SELECT product_id, AVG(star_rating) AS avg_star_rating,
            COUNT(*) AS num
            FROM amazon_reviews_traditional
            GROUP BY product_id) amazon_reviews_traditional
WHERE num >=50
ORDER BY avg_star_rating ASC
LIMIT 1;

```

product_id	avg_star_rating
0072119551	1.18

7. The following query normalizes all comments on that product to lowercase, breaks them into individual words using the SENTENCES function, and passes those to the NGRAMS function to find the five most common bigrams (two-word combinations). Run the query in Hive:

```

SELECT EXPLODE(NGRAMS(SENTENCES(LOWER(review_body))), 2, 5)) AS
bigrams FROM amazon_reviews_traditional WHERE product_id = 0072119551;

```

bigrams
{"ngram": ["this", "book"], "estfrequency": 74.0}
{"ngram": ["the", "exam"], "estfrequency": 54.0}
{"ngram": ["the", "book"], "estfrequency": 34.0}
{"ngram": ["70", "100"], "estfrequency": 27.0}
{"ngram": ["of", "the"], "estfrequency": 21.0}

8. Most of these words are too common to provide much insight. Modify the previous query to find the five most common trigrams (three-word combinations), and then run that query in Hive, which shows the following result:

```

SELECT EXPLODE(NGRAMS(SENTENCES(LOWER(review_body))), 3, 5)) AS
bigrams FROM amazon_reviews_traditional WHERE product_id = 0072119551;

```





```

+-----+
| I just took the 70-100 exam, and while I passed it wasn't because of this book. The exam format is very different from the practice tests in this book & mp; CD--The 70-100 test is now done as case studies.
+-----+
| If you waste your time reading this book in order to study for the 70-100 certification exam, you will be well and truly sorry.<br />The vast majority of this book is concerned with teaching readers about Microsoft's recommended methods of project management and writing white papers. the exam, on the other hand, focuses on case studies, project design (high level), and relational database design.<br />If you need a study guide for this exam, look elsewhere. It is a shame too, as other books in this series have been very useful.
+-----+
2 rows selected (7.651 seconds)

```

The previous step should have displayed two comments:

1. I just took the 70-100 exam, and while I passed it wasn't because of this book. The exam format is very different from the practice tests in this book & CD--The 70-100 test is now done as case studies.
2. If you waste your time reading this book in order to study for the 70-100 certification exam, you will be well and truly sorry. The vast majority of this book is concerned with teaching readers about Microsoft's recommended methods of project management and writing white papers. the exam, on the other hand, focuses on case studies, project design (high level), and relational database design. If you need a study guide for this exam, look elsewhere. It is a shame too, as other books in this series have been very useful.

11. The second comment states that the book's content is irrelevant to the 70-100 exam, unlike similar books in its series. Write and run a query that will display 10 review headlines for product ID 0072119551 in the amazon\_traditional\_reviews table.

```
SELECT review_headline FROM amazon_reviews_traditional WHERE
product_id=0072119551 LIMIT 10;
```

```

+-----+
| review_headline
+-----+
| Wrong material covered for exam
| This Title is Way Off of the Mark
| Covers the wrong material for 70-100
| Has nothing to do with the actual exam
| No need to read this book to pass the exam
| Great information in a totally inaccessible format
| Good Book But Not If You Want To Pass The Exam
| Don't listen to the 4 and 5 star ratings.
| DO NOT BUY THIS BOOK
| DO NOT, I REPEAT, DO NOT PICK UP THIS BOOK FOR A GUIDE
+-----+

```

The query results show that the book's content is good, but the title is misleading. Customers who want a study guide for the exam are purchasing this book based on the title, but the content is not geared towards the exam.

Based on the review\_body and review\_headline columns, it appears that doing text processing has helped this author uncover a title error.

## **STEP 4: Create Hive Queries to Analyze the Sentiment of Data Using Dictionary and Download Data into your PC**

1. Copy the dictionary table, which has **polarity** to show each word's meaning implied as positive or negative, from the main database into the TtoTh Database:

```
CREATE TABLE TtoTH.dictionary AS select * from dictionary
```

```
INFO  : Map 1: 0/1
INFO  : Map 1: 0(+1)/1
INFO  : Map 1: 1/1
INFO  : Moving data to: hdfs://mycluster/apps/hive/warehouse/ttoth.db/dictionary
      from hdfs://mycluster/apps/hive/warehouse/.hive-staging_hive_2018-12-04_02-21-1
1_579_2477824766358443135-1888/-ext-10001
INFO  : Table tooth.dictionary stats: [numFiles=1, numRows=8221, totalSize=30892
2, rawDataSize=300701]
No rows affected (4.921 seconds)
```

We need to use the TtoTH database to query data:

Use TtoTH;

```
0: jdbc:hive2://cis5200-bdcsce-4.compute-6082> use tooth;
No rows affected (0.168 seconds)
```

Make sure that the dictionary table has been created using command:

SHOW tables;

```
0: jdbc:hive2://cis5200-bdcsce-4.compute-6082> show tables;
+-----+
|      tab_name      |
+-----+
| amazon_reviews      |
| amazon_reviews_traditional |
| dictionary           |
+-----+
3 rows selected (0.178 seconds)
0: jdbc:hive2://cis5200-bdcsce-4.compute-6082>
```

2. Using EXPLODE, list all words in review body for product\_id=0072119551, MCSD Analyzing Requirements: Exam 70-100 (MCSD Study Guides), which should produce 3,946 results:

```
SELECT EXPLODE(SPLIT(review_body, ' ')) AS word
FROM amazon_reviews_traditional
WHERE product_id = 0072119551;
```

```
| it
| has
| a
| lot
| of
| Microsoft-specific
| stuff.
| As
| a
| beta
| tester
| for
| the
| exam,
| I
| found
| this
| book
| is
| a
| disappointment
+-----+
3,946 rows selected (173.366 seconds)
```

3. Display words which are accounted for in the Dictionary table and order by polarity, which should display 469 Results:

```
SELECT words, dictionary.polarity
FROM (SELECT explode(split(review_body, ' '))
```

informative	positive	truly	positive
interested	positive	understand	positive
interesting	positive	understand	positive
just	positive	useful	positive
just	positive	useful	positive
just	positive	value	positive
just	positive	vast	positive
just	positive	want	positive
just	positive	want	positive
just	positive	want	positive
just	positive	well	positive
just	positive	well	positive
just	positive	well	positive
just	positive	well	positive
just	positive	white	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	will	positive
just	positive	wink	positive
just	positive	wink	positive
just	positive	wish	positive
just	positive	wish	positive
just	positive	wish	positive
learning	positive	wonder	positive
like	positive	worth	positive
like	positive	worth	positive
live	positive		
luck	positive		

469 rows selected (17.825 seconds)

```
SELECT polarity, sum(TimesWordAppearsInReview)
FROM (SELECT DISTINCT words,
TimesWordAppearsInReview,
polarity
FROM (SELECT words, COUNT(words) as TimesWordAppearsInReview
FROM (SELECT words
```



```

FROM(SELECT explode(split(review_body, ' ')) AS words
FROM amazon_reviews_traditional
WHERE product_id = 0072119551) a join Dictionary on words=word
ORDER BY words) b
GROUP BY words) c
JOIN Dictionary on words=word
ORDER BY words, polarity) d
GROUP by polarity;

```

polarity	_c1
positive	223
negative	114
neutral	145

3 rows selected (17.919 seconds)

5. Even though this book has the lowest average rating, it's "positive" polarity is such a high number. This may be an error, so run the following hive command to determine what positive words may be skewing the results:

```

SELECT DISTINCT c.words, c.TimesWordAppearsInReview, Dictionary.polarity
FROM (SELECT words, COUNT(words) as TimesWordAppearsInReview
FROM(SELECT words FROM(SELECT explode(split(review_body, ' ')) as words
FROM amazon_reviews_traditional
WHERE product_id = 0072119551) a JOIN Dictionary on words=word
ORDER BY words) b
GROUP by words) c JOIN Dictionary on words=word
ORDER by TimesWordAppearsInReview DESC;

```

c.words	c.timeswordappearsinreview	dictionary.polarity
just	39	positive
will	24	positive
help	24	positive
waste	18	negative
good	13	positive
even	11	positive
could	11	neutral
need	9	negative
need	9	neutral
really	9	neutral
content	8	positive
so	8	neutral
real	7	positive
concerning	6	neutral
actual	6	neutral
look	5	neutral
know	5	neutral
very	5	neutral
learn	4	neutral
had	4	negative

6. The words “just” and “will” appear 39 and 24 times respectively. These may or may not be positive in most contexts. We can look at a few comments that contain these words to see if the comment is positive or negative overall:

```
SELECT substr(review_body,0,100)
FROM amazon_reviews_traditional
WHERE review_body like '%just%' and product_id = 0072119551;
```

```

+-----+
| _c0 |
+-----+
| I just took the 70-100 exam, and while I passed it wasn't because of this book. The exam format is  

| I just came back from the 2 hour MCS D 70-100 exam. Not a single question on the exam is covered in  

| This review mostly just for spite to lower rating. Look for my copy of this book on auctions! MSF  

| I give it negative 5 stars. What a fool I was not to have read these reviews earlier. I bought the b  

| It could be argued that this book contains good reference material for Analyzing Requirements. Argu  

| I've just passed the exam. This book is not useful for passing exam 70-100. If you buy it for the ex  

| Syngress should be shot, along with the folks who reviewed and approved the content for logo use. I  

| Having just taken Exam 70-100, I can say that this book does NOT prepare you for it.<br />The book d  

| Having read the book and taken the test, I must say that this book does a good job covering the MSF,  

| This book mainly covers MSF. If you are interested in MSF then it might be worth buying. I just took  

| This is a great book if you want to learn about the Microsoft Solutions Framework (MSF). I just wrot  

| just did the exam.<br />This is a good book, and after you ignore the first 5 chapters it DOES fol  

| Having just flunked the exam miserably after studying all 3 books currently available, only one thre  

+-----+
13 rows selected (16.154 seconds)
0: jdbc:hive2://cis5200-bdcsce-4.compute-6082>

```



7. We can see from the comments that they are not positive overall. The words “just” and “will” are skewing our results. We need to clean up the data by omitting these words by using the following HIVE command:

```
CREATE TABLE TtoTH.dictionary_adj AS select * from dictionary where word not in ('just','will');
```

```
INFO  : Map 1: 0/1
INFO  : Map 1: 0(+1)/1
INFO  : Map 1: 1/1
INFO  : Moving data to: hdfs://mycluster/apps/hive/warehouse/ttoth.db/dictionary_adj from hdfs://mycluster/apps/hive/warehouse/ttoth.db/.hive-staging_hive_2018-12-04_02-34-40_631_711012909501973834-1888/-ext-10001
INFO  : Table ttotth.dictionary_adj stats: [numFiles=1, numRows=8216, totalSize=308756, rawDataSize=300540]
No rows affected (4.49 seconds)
```

8. Show totals by polarity using the new dictionary without words “just” and “will”

```
SELECT polarity, SUM(TimesWordAppearsInReview)
      FROM (SELECT DISTINCT words, TimesWordAppearsInReview, polarity
            FROM (SELECT words, COUNT(words) as TimesWordAppearsInReview
                  FROM (SELECT words
                        FROM (SELECT explode(split(review_body, ' ')) AS words
                        FROM amazon_reviews_traditional
                        WHERE product_id = 0072119551) a
                        JOIN dictionary_adj on words=word
                        ORDER BY words) b
                        GROUP BY words) c
                        JOIN dictionary_adj on words=word
                        ORDER BY words, polarity) d
                        GROUP BY polarity;
```

```

INFO  : Map 1: 5/5      Map 6: 1/1      Map 7: 1/1      Reducer 2: 1/1 Reducer
3: 8/8 Reducer 4: 1/1 Reducer 5: 3(+1)/4
INFO  : Map 1: 5/5      Map 6: 1/1      Map 7: 1/1      Reducer 2: 1/1 Reducer
3: 8/8 Reducer 4: 1/1 Reducer 5: 4/4
+-----+-----+
| polarity | _c1 |
+-----+-----+
| positive | 160 |
| negative | 114 |
| neutral  | 145 |
+-----+-----+
3 rows selected (17.855 seconds)

```

9. Now that we've cleaned our data we can get around to computing the sentiment. Use the following 3 Hive commands to create 3 views that will allow us to do that:

```

CREATE view IF NOT EXISTS temp_One AS
    SELECT product_id, words
FROM amazon_reviews_traditional
    lateral view EXPLODE(SENTENCES(LOWER(review_body))) dummy AS words;

```

```

CREATE view IF NOT EXISTS temp_Two AS
    SELECT product_id, word
FROM temp_One
    lateral view explode( words ) dummy AS word;

```

```

CREATE view IF NOT EXISTS temp_Three AS
    SELECT product_id,
    temp_two.word,
    case d.polarity
    when 'negative' then -1
    when 'positive' then 1
    else 0 end as polarity
    from temp_two left outer join dictionary_adj d on temp_two.word = d.word;

```

temp_three.product_id	temp_three.word	temp_three.polarity
product_id	review_body	0
0385730586	this	0
0385730586	boook	0
0385730586	was	0
0385730586	a	0
0385730586	great	1
0385730586	one	0
0385730586	that	0
0385730586	you	0
0385730586	could	0

```
0: jdbc:hive2://cis5200-bdcsce-4.compute-6082> CREATE view IF NOT EXISTS temp_One
AS SELECT product_id, words FROM amazon_reviews_traditional lateral view EXPLO
DE(SENTENCES(LOWER(review_body))) dummy AS words; CREATE view IF NOT EXISTS temp
_Two AS SELECT product_id, word FROM temp_One lateral view explode( words ) dumm
y AS word; CREATE view IF NOT EXISTS temp_Three AS SELECT product_id, temp_two.w
ord, case d.polarity when 'negative' then -1 when 'positive' then 1 else 0 end as
s polarity from temp_two left outer join dictionary_adj d on temp_two.word = d.w
ord;
No rows affected (0.22 seconds)
No rows affected (0.211 seconds)
No rows affected (0.242 seconds)
```

We can determine overall sentiment of each product id using average sentiment:

```
CREATE View IF NOT EXISTS productsreview_sentiment AS
  SELECT product_id,
  CASE
    when sum( polarity ) > 0 then 'positive'
    when sum( polarity ) < 0 then 'negative'
    ELSE 'neutral' end as sentiment, sum( polarity ) as sentiment_rating,
  AVG(polarity)
  AS sentiment_average
FROM temp_three
GROUP by product_id;
```

```
0: jdbc:hive2://cis5200-bdcsce-4.compute-6082> CREATE View IF NOT EXISTS product
sreview_sentiment AS SELECT product_id, CASE when sum( polarity ) > 0 then 'posi
tive' when sum( polarity ) < 0 then 'negative' ELSE 'neutral' end as sentiment,
sum( polarity ) as sentiment_rating, AVG(polarity) AS sentiment_average FROM tem
p_three GROUP by product_id;
No rows affected (0.24 seconds)
```

10. You have to query data from productreview\_sentiment to see if it has the correct data and values:

```
SELECT * from productsreview_sentiment LIMIT 10;
```

product_id	sentiment	sentiment_rating	sentiment_average
0001527355	positive	21	0.11229946524064172
0002200155	positive	10	0.1388888888888889
0002151928	positive	1	0.01694915254237288
0002250381	positive	3	0.1
0001983679	positive	8	0.0761904761904762
0002161621	positive	10	0.046296296296296294
0001006002	positive	6	0.09230769230769231
0001857029	positive	11	0.07857142857142857
0002000172	positive	3	0.036585365853658534
000215725X	positive	38	0.031198686371100164

11. Create Star Rating view:

```
CREATE View IF NOT EXISTS productsreview_avg_star_rating AS
    SELECT product_id, FORMAT_NUMBER(avg_star_rating,2) AS avg_star_rating
    FROM (SELECT product_id, AVG(star_rating) AS avg_star_rating,
        COUNT(*) AS num
    FROM amazon_reviews_traditional
    GROUP by product_id) amazon_reviews_traditional
WHERE num >=100;
```

```
0: jdbc:hive2://cis5200-bdcscce-4.compute-6082> CREATE View IF NOT EXISTS product
sreview_avg_star_rating AS SELECT product_id, FORMAT_NUMBER(avg_star_rating,2) A
S avg_star_rating FROM (SELECT product_id, AVG(star_rating) AS avg_star_rating,
COUNT(*) AS num FROM amazon_reviews_traditional GROUP by product_id) amazon_revi
ews_traditional WHERE num >=100;
No rows affected (0.225 seconds)
```

12. Consolidate the Star Ratings and Sentiment Information:

```
CREATE View IF NOT EXISTS
    consolidates_sentiment_starrating AS
    SELECT b.product_id, b.sentiment,
    FORMAT_NUMBER(b.sentiment_average,4) as Sentiment_Range,
    a.avg_star_rating
```

FROM productsreview\_avg\_star\_rating a LEFT OUTER JOIN  
productsreview\_sentiment b on a.product\_id = b.product\_id;

```
0: jdbc:hive2://cis5200-bdcsce-4.compute-6082> CREATE View IF NOT EXISTS consoli
dates_sentiment_starrating AS SELECT b.product_id, b.sentiment, FORMAT_NUMBER(b.
sentiment_average,4) as Sentiment_Range, a.avg_star_rating FROM productsreview_a
vg_star_rating a LEFT OUTER JOIN productsreview_sentiment b on a.product_id = b.
product_id;
No rows affected (0.276 seconds)
```

consolidates_sentiment_starrating.product_id	consolidates_sentiment_starrating.sentiment	consolidates_sentiment_starrating.sentiment_range	consolidates_sentiment_starrating.avg_star_rating
0060275103	positive	0.0640	4.81
0061015725	positive	0.0125	3.36
0112187459	positive	0.0478	4.46
0345335911	positive	0.0438	4.12
0345378482	positive	0.0076	4.14
0380814676	positive	0.0299	4.45
0385335482	positive	0.0354	4.35
038540981X	positive	0.0210	4.16
0385403622	positive	0.0369	4.68
0385501560	positive	0.0160	4.07

### 13. Create Product Name and Product Table View

CREATE View IF NOT EXISTS ProductTitle AS  
SELECT DISTINCT product\_title, product\_id  
FROM amazon\_reviews\_traditional;

```
0: jdbc:hive2://cis5200-bdcsce-4.compute-6082> CREATE View IF NOT EXISTS Product
Title AS SELECT DISTINCT product_title, product_id FROM amazon_reviews_tradition
al;
No rows affected (0.226 seconds)
```

### 14. Create Table with all of our sentiment information:

CREATE table IF NOT EXISTS AmazonReviewsInfo  
STORED AS orc AS  
SELECT b.product\_title,a.product\_id,a.sentiment, a.Sentiment\_Range,  
a.avg\_star\_rating  
FROM consolidates\_sentiment\_starrating a LEFT OUTER JOIN ProductTitle b  
on a.product\_id=b.product\_id;



```

OpenSSH SSH client
jdbc:hive2://cis5200-bdscse-4.compute-6082: create table IF NOT EXISTS AmazonReviewsInfo
jdbc:hive2://cis5200-bdscse-4.compute-6082: stored as orc as
jdbc:hive2://cis5200-bdscse-4.compute-6082: select b.product_title,a.product_id,a.sentiment, a.Sentiment_Range, a.avg_star_rating
jdbc:hive2://cis5200-bdscse-4.compute-6082: from consolidates_sentiment_starrating a LEFT OUTER JOIN ProductTitle b on a.product_id=b.product_id;

INFO : Tez session hasn't been created yet. Opening session
INFO : Dag name: create table IF NOT EXISTS product_id(Stage-1)
INFO :
INFO : Status: Running (Executing on YARN cluster with App id application_1541708221620_0486)

INFO : Map 1: -/- Map 5: -/- Map 7: -/- Map 8: -/- Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 0/40 Map 5: 0/40 Map 7: 0/1 Map 8: 0/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 0(+1)/40 Map 5: 0/40 Map 7: 0(+1)/1 Map 8: 0/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 0(+2)/40 Map 5: 0/40 Map 7: 0(+1)/1 Map 8: 0/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 0(+3)/40 Map 5: 0/40 Map 7: 0(+1)/1 Map 8: 0/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 0(+3)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 0(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 0(+3)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 0(+2)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 0(+3)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 1(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 1(+3)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 1(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 1(+4)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 1(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 2(+3)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 1(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 3(+2)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 1(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 4(+2)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 1(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 4(+2)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 2(+2)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 4(+4)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 3(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 5(+4)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 4(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 7(+2)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 4(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 8(+5)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 4(+5)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 8(+1)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 5(+3)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 8(+3)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 6(+2)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 8(+4)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 8(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 9(+4)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 8(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 10(+3)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 8(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 12(+1)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 9(+3)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 12(+2)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 9(+3)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 13(+1)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 9(+4)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 13(+1)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 12(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 14(+0)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 12(+2)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 14(+2)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 12(+2)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 14(+3)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 12(+2)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49
INFO : Map 1: 14(+4)/40 Map 5: 0/40 Map 7: 1/1 Map 8: 13(+1)/40 Reducer 2: 0/49 Reducer 3: 0/115 Reducer 4: 0/141 Reducer 6: 0/213 Reducer 9: 0/49

```

15. You have to query data from the table to see if it has the correct data and values:

SELECT product\_id, sentiment, Sentiment\_Range, avg\_star\_rating  
FROM AmazonReviewsInfo order by sentiment\_range;

```

jdbc:hive2://cis5200-bdscse-4.compute-6082: select product_title, product_id, sentiment, Sentiment_Range, avg_star_rating
jdbc:hive2://cis5200-bdscse-4.compute-6082: from AmazonReviewsInfo order by sentiment_range LIMIT 10;

INFO : Session is already open
INFO : Dag name: select product_title, product_id, senti...(Stage-1)
INFO :
INFO : Status: Running (Executing on YARN cluster with App id application_1541708221620_0490)

INFO : Map 1: 0/1 Reducer 2: 0/1
INFO : Map 1: 0(+1)/1 Reducer 2: 0/1
INFO : Map 1: 1/1 Reducer 2: 0/1
INFO : Map 1: 1/1 Reducer 2: 0(+1)/1
INFO : Map 1: 1/1 Reducer 2: 1/1

product_title | product_id | sentiment | sentiment_range | avg_star_rating |
-----|-----|-----|-----|-----|
9-11 | 1583224890 | negative | -0.0004 | 3.76 |
American Psycho | 0679735771 | negative | -0.0004 | 3.52 |
1st to Die: A Novel (Women's Murder Club, No 1) | 0446610038 | negative | -0.0005 | 3.55 |
The Hunting of the President: The Ten-Year Campaign to Destroy Bill and Hillary Clinton | 0312245475 | negative | -0.0005 | 4.55 |
Pop Goes the Weasel (Alex Cross) | 0446608815 | negative | -0.0005 | 3.45 |
Will They Ever Trust Us Again?: Letters From the War Zone | 0743271521 | negative | -0.0006 | 3.82 |
Gods and Generals: A Novel of the Civil War (Civil War Trilogy) | 0345422473 | negative | -0.0006 | 3.96 |
Isle of Dogs (Andy Brazil) | 0425182908 | negative | -0.0006 | 1.46 |
The Chamber | 0440220602 | negative | -0.0007 | 3.58 |
SSN | 0425173534 | negative | -0.0007 | 1.93 |

10 rows selected (6.482 seconds)
jdbc:hive2://cis5200-bdscse-4.compute-6082>

```

16. Run the following shell commands to make sure that the directory tmp/data/info is there and that beeline command will work:

```
hdfs dfs -mkdir tmp/data/info
hdfs dfs -chmod -R o+w tmp/
```

```
drwxr-xrwx mespino6 hdfs 0 2018-12-04 03:19 /user/respino6/tmp/data
-bash-4.1$ hdfs dfs -mkdir tmp/data/info
-bash-4.1$ hdfs dfs -chmod -R o+w tmp/
```

17. Download the file to HDFS path"/user/respino6/tmp/data/info":

```
CREATE TABLE IF NOT EXISTS AmazonReviewsInformation
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ","
STORED AS TEXTFILE
LOCATION "/user/respino6/tmp/data/info"
AS
select product_id, sentiment, Sentiment_Range, avg_star_rating
from AmazonReviewsInfo order by sentiment_range;
```

```
OpenSSH SSH client
jdbc:hive2://cis5200-bdcsce-4.compute-6082 CREATE TABLE IF NOT EXISTS AmazonReviewsInformation
jdbc:hive2://cis5200-bdcsce-4.compute-6082 ROW FORMAT DELIMITED
jdbc:hive2://cis5200-bdcsce-4.compute-6082 FIELDS TERMINATED BY ","
jdbc:hive2://cis5200-bdcsce-4.compute-6082 STORED AS TEXTFILE
jdbc:hive2://cis5200-bdcsce-4.compute-6082 LOCATION "/user/respino6/tmp/data/info"
jdbc:hive2://cis5200-bdcsce-4.compute-6082 AS
jdbc:hive2://cis5200-bdcsce-4.compute-6082 select product_id, sentiment, Sentiment_Range, avg_star_rating
jdbc:hive2://cis5200-bdcsce-4.compute-6082 from AmazonReviewsInfo order by sentiment_range;
INFO : Session is already open
INFO : Dag name: CREATE TABLE IF NOT EXISTS...sentiment_range(Stage-1)
INFO :
INFO : Status: Running (Executing on YARN cluster with App id application_1541708221620_0486)
INFO : Map 1: -/- Reducer 2: 0/1
INFO : Map 1: 0/1 Reducer 2: 0/1
INFO : Map 1: 0(+1)/1 Reducer 2: 0/1
INFO : Map 1: 0(+1)/1 Reducer 2: 0/1
INFO : Map 1: 1/1 Reducer 2: 0/1
INFO : Map 1: 1/1 Reducer 2: 0(+1)/1
INFO : Map 1: 1/1 Reducer 2: 1/1
INFO : Moving data to: /user/respino6/tmp/data/info from hdfs://mycluster/apps/hive/warehouse/ttoth.db/.hive-staging_hive_2018-12-04_02-17-44_628_6625295011458962946-1888/-ext-
INFO : Table tttoth.amazonreviewsinformation stats: [numFiles=1, numRows=1727, totalSize=55325, rawDataSize=53598]
INFO : 10 rows affected (0.102 seconds)
No rows affected (0.102 seconds)
```

18. You have to query data from the table to see if it has the correct data and values:

```
SELECT* from AmazonReviewsInfo LIMIT 10;
```

```
jdbc:hive2://cis5200-bdcsce-4.compute-6082 select * from AmazonReviewsInformation LIMIT 10;
amazonreviewsinformation.product_id | amazonreviewsinformation.sentiment | amazonreviewsinformation.sentiment_range | amazonreviewsinformation.avg_star_rating |
0679735771 | negative | -0.0004 | 3.52 |
1583224890 | negative | -0.0004 | 3.76 |
0312245475 | negative | -0.0005 | 4.55 |
0446610038 | negative | -0.0005 | 3.55 |
0446608815 | negative | -0.0005 | 3.45 |
0425182908 | negative | -0.0006 | 1.46 |
0743271521 | negative | -0.0006 | 3.82 |
0345422473 | negative | -0.0006 | 3.96 |
0812504798 | negative | -0.0007 | 3.40 |
0440220602 | negative | -0.0007 | 3.58 |
10 rows selected (0.102 seconds)
jdbc:hive2://cis5200-bdcsce-4.compute-6082>
```



Open another terminal with git bash, minty, or putty, which is to connect the Oracle Cloud to download the output file 000000\_0 at the HDFS path `"/user/mespino6/tmp/data/info"`:

19.

```
hdfs dfs -ls /user/mespino6/tmp/data/info
```

20.

```
hdfs dfs -get /user/mespino6/tmp/data/info/00000* 0
```

21. ls -al

```

OpenSSH SSH client
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Downloads>ssh mespino6@129.150.128.177
mespino6@129.150.128.177's password:
-bash-4.1$ hdfs dfs -ls /user/respino6/tmp/data/info
-bash-4.1$ hdfs dfs -ls /user/respino6/tmp/data/info
-bash-4.1$ hdfs dfs -ls /user/respino6/tmp/data
Found 2 items
drwxr-xrwx - mespino6 hdfs 0 2018-12-04 00:31 /user/respino6/tmp/data/info
drwxr-xrwx - mespino6 hdfs 0 2018-12-01 01:25 /user/respino6/tmp/data/tables
-bash-4.1$ hdfs dfs -ls /user/respino6/tmp/data/info
Found 1 items
-rwxr-xrwx 2 hdfsce_admin hdfs 55325 2018-12-04 02:17 /user/respino6/tmp/data/info/000000_0
-bash-4.1$ hdfs dfs -get /user/respino6/tmp/data/info/000000* 0
-bash-4.1$ ls -al
total 1298532
drwx----- 6 mespino6 mespino6 4096 Dec 4 02:26 .
drwxr-xr-x 33 root root 4096 Nov 7 19:47 ..
-rw-r--r-- 1 mespino6 mespino6 55325 Dec 4 02:26 000000_0
-rw-rw-r-- 1 mespino6 mespino6 1329539135 Nov 24 2017 amazon_reviews_us_Books_v1_02.tsv.gz
-rw----- 1 mespino6 mespino6 32469 Dec 4 00:45 .bash_history
drwxrwxr-x 2 mespino6 mespino6 4096 Sep 26 03:12 .booktime
-rw-rw-r-- 1 mespino6 mespino6 5894 Nov 30 22:08 .hivehistory
drwxrwxr-x 3 mespino6 mespino6 4096 Sep 8 2016 .hadoop
drwxrwxr-x 2 mespino6 mespino6 4096 Sep 26 02:16 .oracle_fw_usage
-rw-rw-r-- 1 mespino6 mespino6 20667 Nov 30 22:05 .pig_history
drwx----- 2 mespino6 mespino6 4096 Sep 26 03:17 .ssh
-bash-4.1$

```

22. For Windows user, you may use psftp to download the file. You need to download it at <http://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe>. In order to download 000000 0, you have to run psftp as follows:

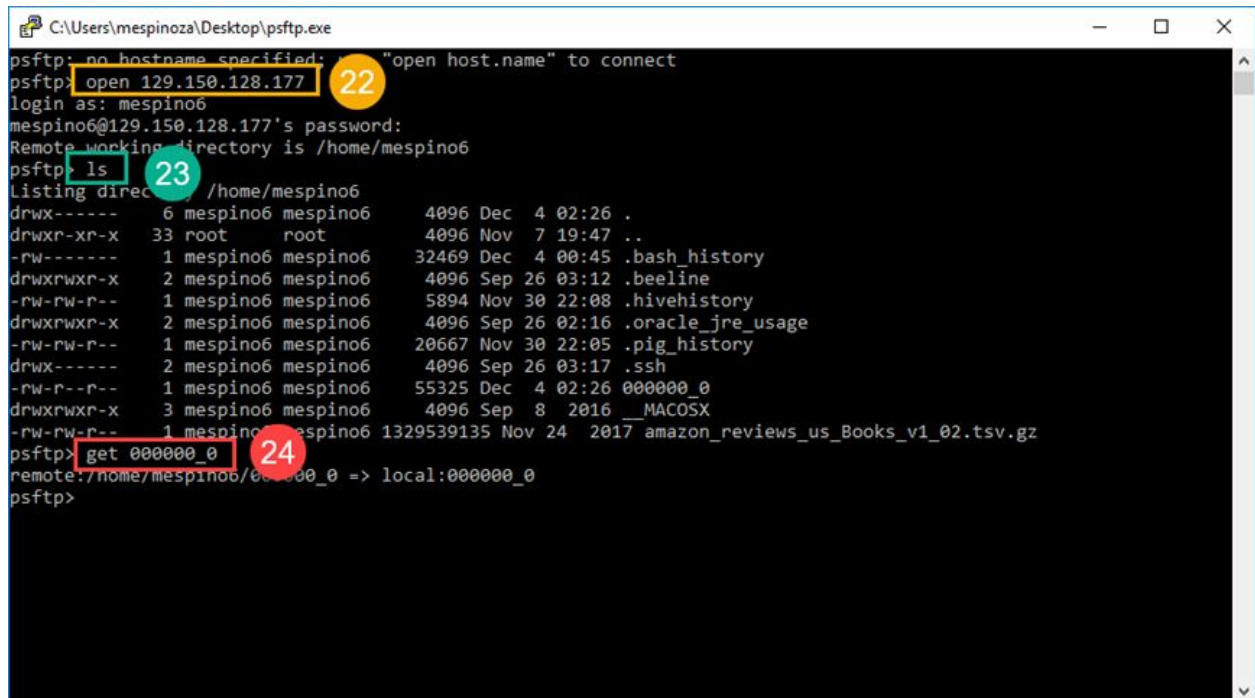
open 129.150.128.177

23. List directory to make sure you are in the right place:

ls

24. Download the file 000000\_0 using get command:

get 000000\_0

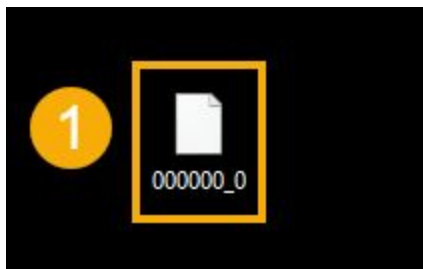


The screenshot shows a terminal window titled "C:\Users\mespinoza\Desktop\psftp.exe". The user enters the command "open 129.150.128.177" (highlighted with a yellow box and a yellow circle with the number 22). The terminal prompts for a login as "mespino6" and a password. It then shows the remote working directory is "/home/mespino6". The user enters "ls" (highlighted with a green box and a green circle with the number 23), and the terminal displays a directory listing. The user then enters "get 000000\_0" (highlighted with a red box and a red circle with the number 24), and the terminal shows the file being downloaded from the remote directory to the local machine.

```
C:\Users\mespinoza\Desktop\psftp.exe
psftp> no hostname specified: "open host.name" to connect
psftp> open 129.150.128.177
login as: mespino6
mespino6@129.150.128.177's password:
Remote working directory is /home/mespino6
psftp> ls
Listing directory /home/mespino6
drwx----- 6 mespino6 mespino6 4096 Dec 4 02:26 .
drwxr-xr-x 33 root root 4096 Nov 7 19:47 ..
-rw----- 1 mespino6 mespino6 32469 Dec 4 00:45 .bash_history
drwxrwxr-x 2 mespino6 mespino6 4096 Sep 26 03:12 .beeline
-rw-rw-r-- 1 mespino6 mespino6 5894 Nov 30 22:08 .hivehistory
drwxrwxr-x 2 mespino6 mespino6 4096 Sep 26 02:16 .oracle_jre_usage
-rw-rw-r-- 1 mespino6 mespino6 20667 Nov 30 22:05 .pig_history
drwx----- 2 mespino6 mespino6 4096 Sep 26 03:17 .ssh
-rw-r--r-- 1 mespino6 mespino6 55325 Dec 4 02:26 000000_0
drwxrwxr-x 3 mespino6 mespino6 4096 Sep 8 2016 __MACOSX
-rw-rw-r-- 1 mespino6 mespino6 1329539135 Nov 24 2017 amazon_reviews_us_Books_v1_02.tsv.gz
psftp> get 000000_0
remote:/home/mespino6/000000_0 => local:000000_0
psftp>
```

## STEP 5: Loading Data Into Power BI

1. You have to open the following file in Excel using the Text Import Wizard.



2. You have to specify Comma as a Delimiter.

2

Text Import Wizard - Step 2 of 3

This screen lets you set the delimiters your data contains. You can see how your text is affected in the preview below.

**Delimiters**

☐ Tab

☐ Semicolon

☒ Comma

☐ Space

☐ Other:

☐ Treat consecutive delimiters as one

Text qualifier:

**Data preview**

0679735771	negative	-0.0004	3.52
1583224890	negative	-0.0004	3.76
0312245475	negative	-0.0005	4.55
0446610038	negative	-0.0005	3.55
0446608815	negative	-0.0005	3.45

Cancel < Back Next > Finish

3. For the first row of the file, you need to insert the header to each column as follows: Product\_ID sentiment sentiment\_range avg\_star\_rating

000000\_0 - Excel

FILE HOME INSERT New Tab PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER

Clipboard Font Alignment Number

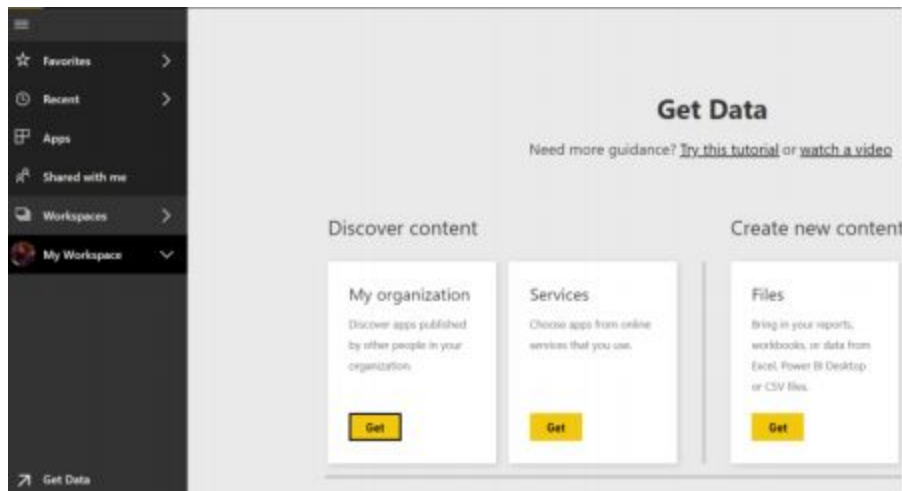
D1 : avg\_star\_rating

	A	B	C	D	E	F	G	H	I	J
1	Product_ID	sentiment	sentiment_range	avg_star_rating						
2	0679735771	negative	-0.0004	3.52						
3	1583224890	negative	-0.0004	3.76						
4	312245475	negative	-0.0005	4.55						
5	446610038	negative	-0.0005	3.55						
6	446608815	negative	-0.0005	3.45						
7	425182908	negative	-0.0006	1.46						
8	743271521	negative	-0.0006	3.82						
9	345422473	negative	-0.0006	3.96						
10	812504798	negative	-0.0007	3.4						

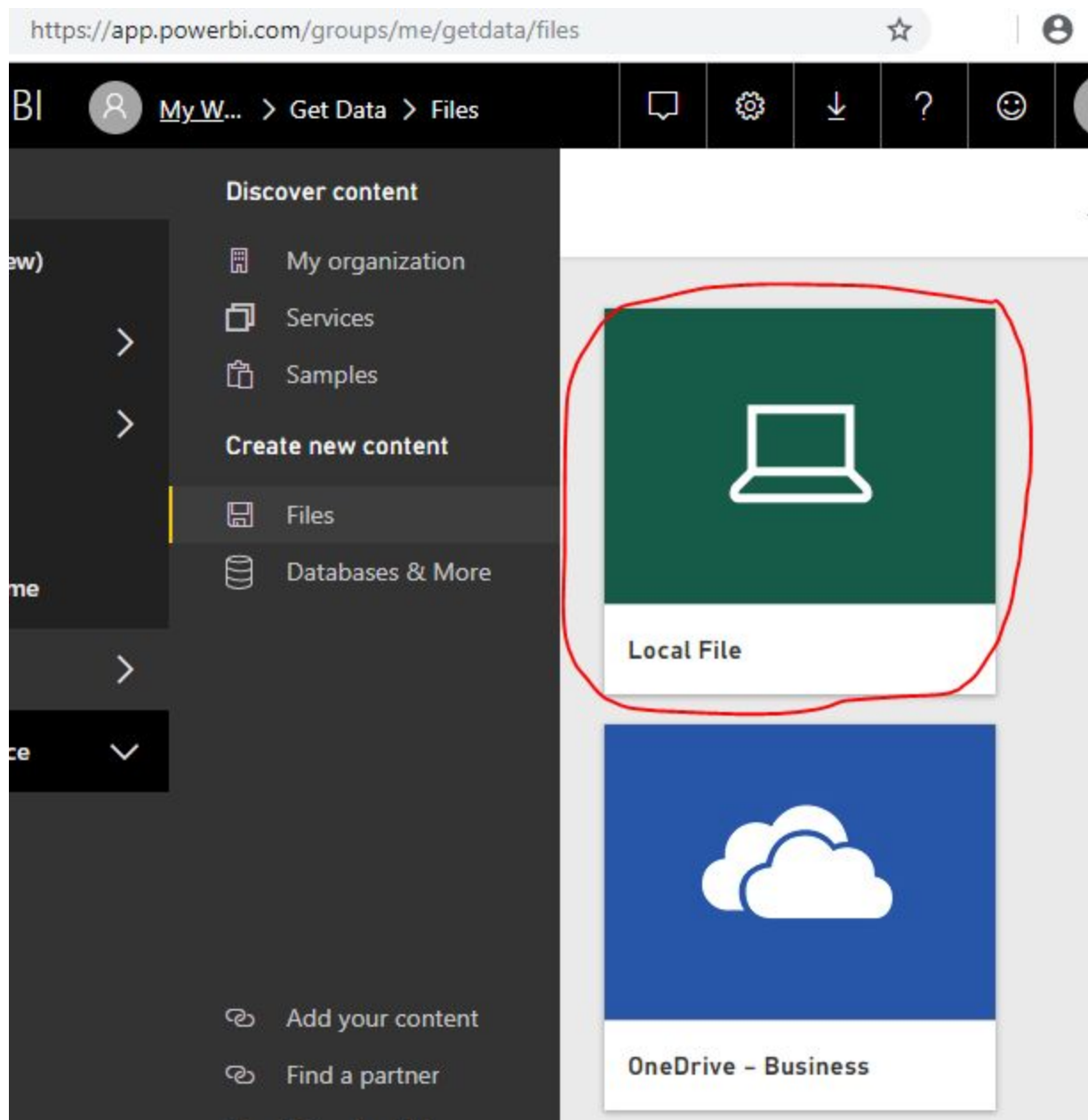
4. Then you have to save the file as comma separated value format, that is, as 000000\_0.csv

5. You need to sign in Power BI website using your school account to import the result data into Power BI to visually explore the data.

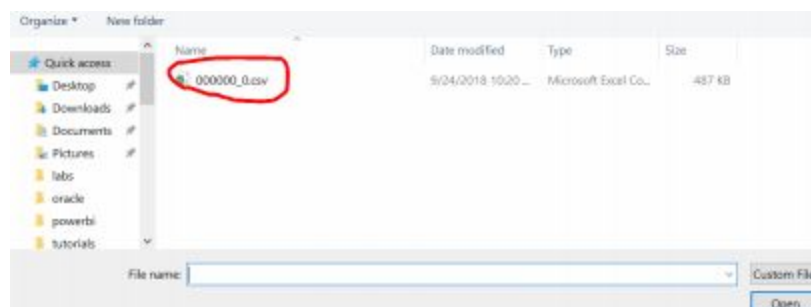
6. Open a web browser and go to sign in with your school account at:  
<https://app.powerbi.com>



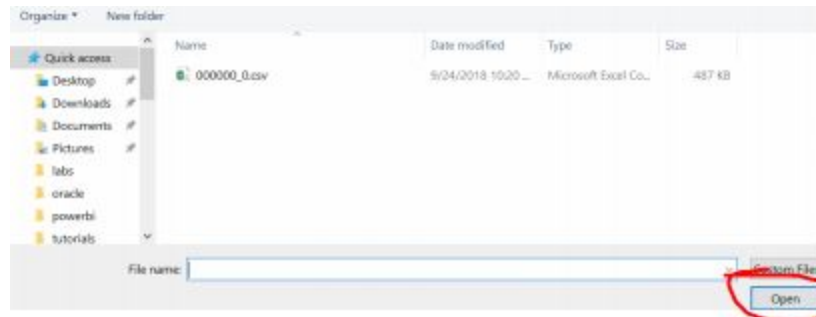
7. Once you sign in, you will see the following pages. Select Local File to upload your output file "000000\_0.csv"



8. You will see the following window popped up:



9. Select the file and open it:



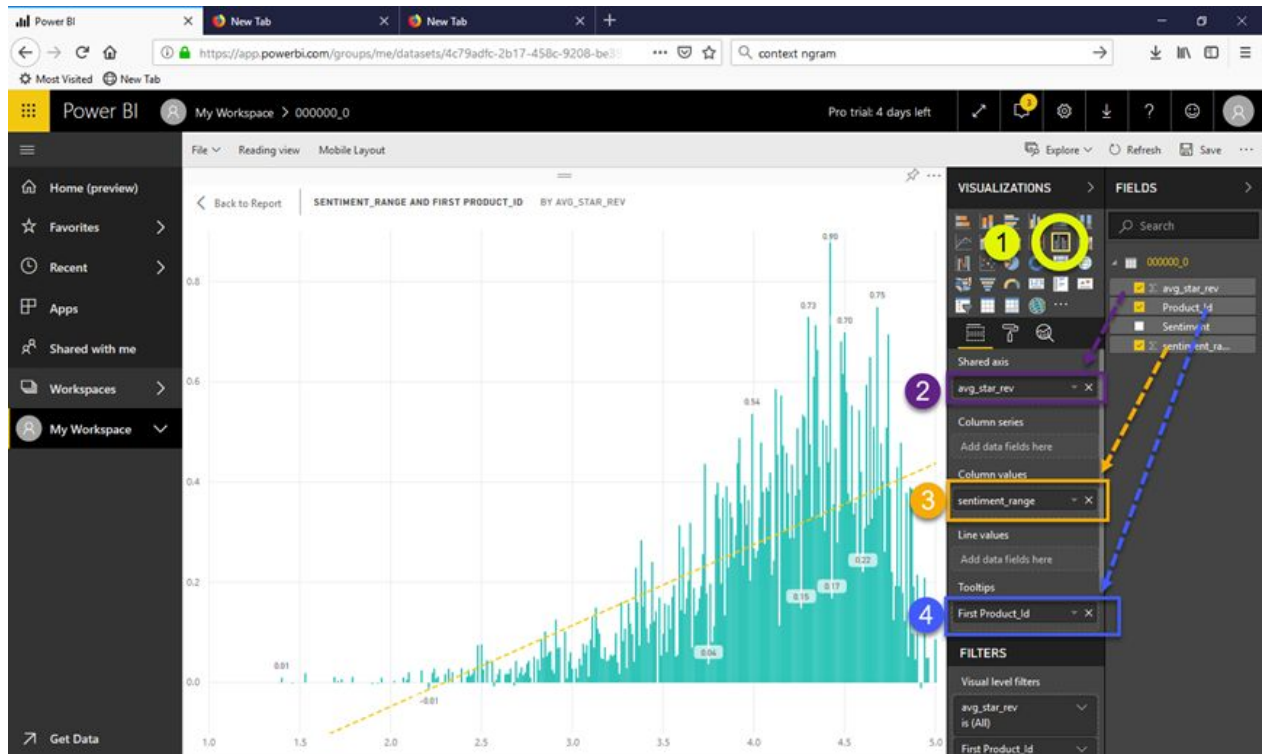
10. Now you will see the following page at your Power BI. Select “View dataset”:



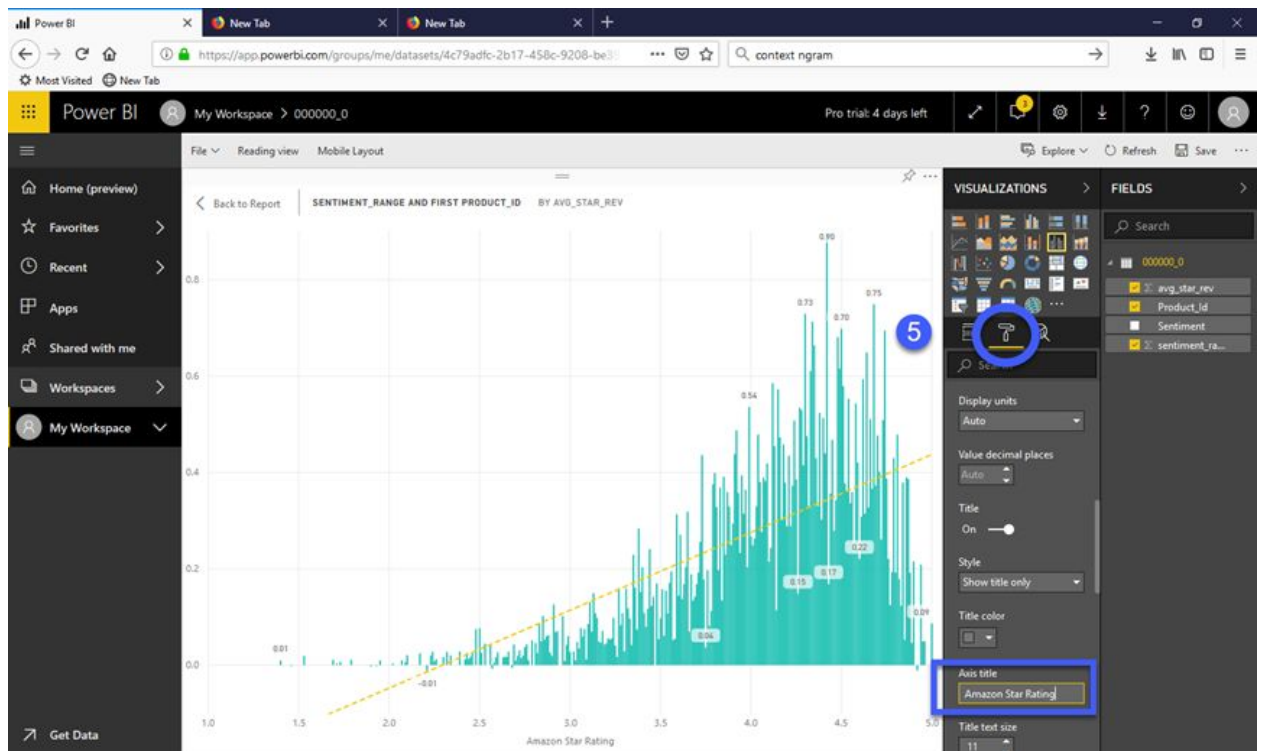
## STEP 6: Visualizing Data In Power BI

1. Select graph
2. Drag avg\_star\_rating under shared axis
3. Drag sentiment\_range under Column values
4. Drag Product\_Id under Tooltips



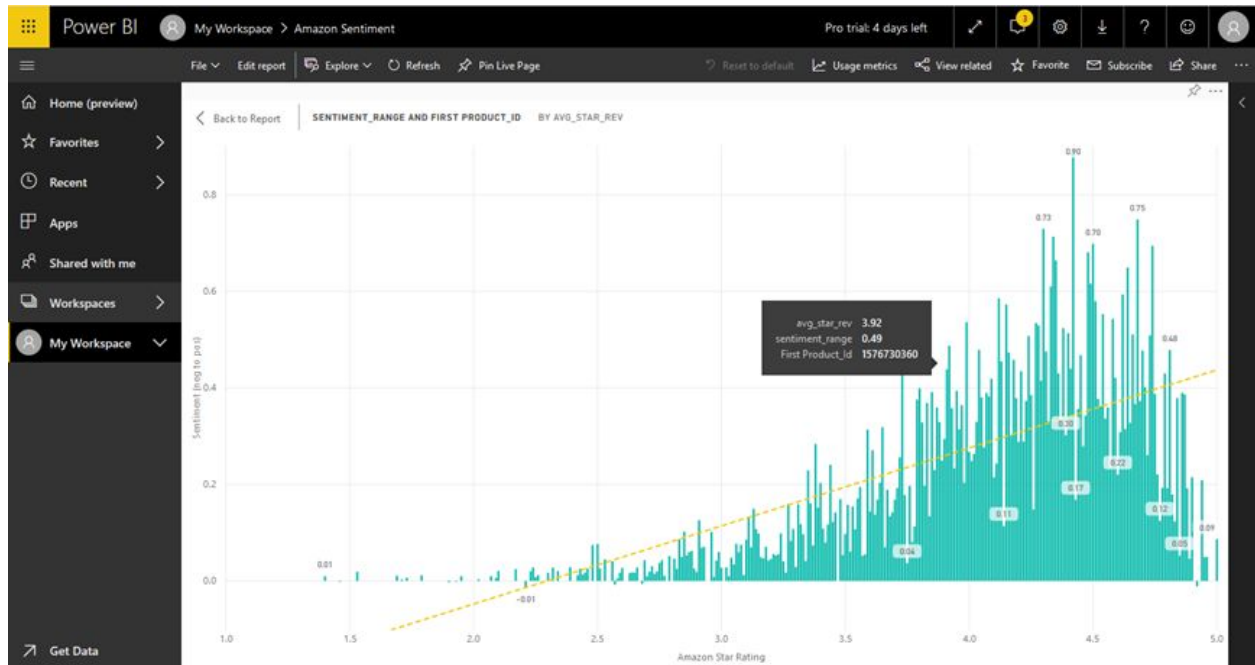


5. Go to design tab and change Axis title to Amazon Star Rating



6. Hide the Visualizations options and you have your bar graph displayed in full!





Github Link:  
<https://github.com/ttothcalstate/Final>