



# CIS 5200 Term Project Tutorial



**Authors:** [Jaydeep J Chopde](#) ; [Maitri S Shah](#);

[Monika Mishra](#); [Pankti N Parikh](#)

[Rakshith Chandan Babu](#)

**Instructor:** [Jongwook Woo](#)

**Date:** 12/15/2018

## Amazon Product Review Data Analysis

### Term Project Group 2

---

#### Objectives

In this lab you will analyze and visualize Amazon Product review data. Thus,

- You should learn how to download Amazon Product review data to the local systems in Oracle Cloud.
- Then, you will learn how to upload it to HDFS.
- You will figure out how to manipulate and analyze the data in HDFS using HiveQL.
- You will also practice how to visualize the result in Excel, Tableau and Powerbi.

#### Introduction

Customer on any online shopping sites make purchasing decisions based on reviews and ratings. And so, it's very important business to know various shopping patterns with a review dataset. Here you will do data analysis based on Amazon product review.

# Platform Specification

Cluster Version – Oracle Big Data Compute Edition

Number of Nodes – 5

Memory size – 150 GB

CPU – 20 vCPU

CPU speed – 2.20 GHz

HDFS capacity – 147 GB

Storage – 678 GB

## Prerequisites

Everything you need to go through the scripts and queries is already provisioned with the cluster. To export the analyzed data to Microsoft Excel, you must meet the following requirements:

- You must have an ip address to connect to Oracle Cloud .
- You must have **Microsoft Excel 2010, 2013 or 2016** installed.
- You must have Tableau installed on your machine for visualizations.
- You must have an account with the PowerBI.

## 1. Connect to Oracle Cloud: Big Data Compute

You need to remotely access your Oracle Big Data that you executed in your Oracle Cloud account using *ssh*. For example, for the user name and ip address: **mmishra2**, you need to run the following with the ip address given:

```
$ ssh mmishra2@ipaddress
```

When asked for password, type in your user name again and enter.

```
Last login: Thu Nov 15 12:48:21 on console
[Monikas-MBP:~ monikamishra$ ssh mmishra2@129.150.205.68
[mmishra2@129.150.205.68's password:
-bash-4.1$ █
```

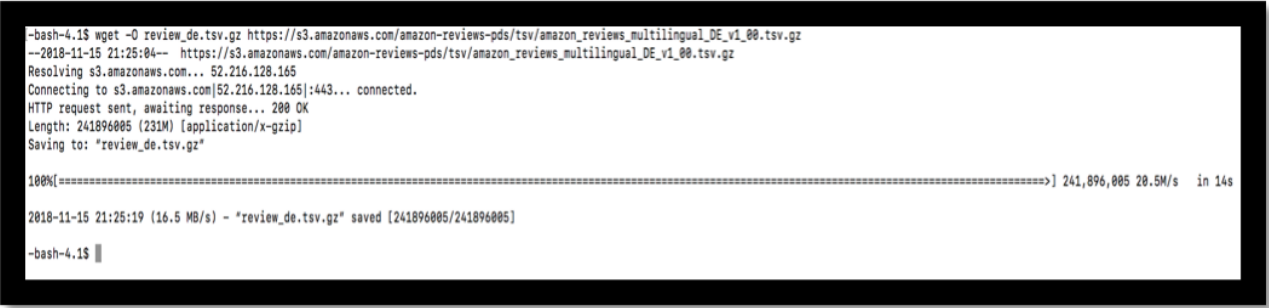
You are now connected to Oracle cloud.

## 2. Amazon review data loaded into Oracle Big Data

Below is the location of the Amazon product review data that is used for this sample. You can download the data file from amazonaws:

```
$ wget -O review_de.tsv.gz https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_DE_v1_00.tsv.gz
```

You should get something like this:



```
-bash-4.1$ wget -O review_de.tsv.gz https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_DE_v1_00.tsv.gz
--2018-11-15 21:25:04-- https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_DE_v1_00.tsv.gz
Resolving s3.amazonaws.com... 52.216.128.165
Connecting to s3.amazonaws.com[52.216.128.165]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 241896005 (231M) [application/x-gzip]
Saving to: "review_de.tsv.gz"

100%[=====] 241,896,005 28.5M/s in 14s

2018-11-15 21:25:19 (16.5 MB/s) - "review_de.tsv.gz" saved [241896005/241896005]

-bash-4.1$
```

Similarly download other files as well.

```
$ wget -O review_us.tsv.gz https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_US_v1_00.tsv.gz

$ wget -O review_uk.tsv.gz https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_UK_v1_00.tsv.gz

$ wget -O review_fr.tsv.gz https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_FR_v1_00.tsv.gz
```

Then, you need to unzip the files.

```
$ gunzip review_us.tsv.gz
$ gunzip review_uk.tsv.gz
$ gunzip review_de.tsv.gz
$ gunzip review_fr.tsv.gz
```

You also need to download the US and Germany and French dictionary files using command as below:

```
$ wget -O dictionary_us.tsv https://s3.amazonaws.com/hipicdatasets/dictionary.tsv
$ wget -O dictionary_germany.zip
http://www.ulliwaltinger.de/sentiment/GermanPolarityClues-
2012.zip

$ wget -O dictionary_fr.csv http://advanse.lirmm.fr/FEEL.csv
```

Then, you need to unzip dictionary\_germany.zip file and you will see 6 files are uncompressed under “GermanPolarityClues-2012” directory:

```
$ unzip dictionary_germany.zip
$ ls GermanPolarityClues-2012
```

```
l-bash-4.1$ ls GermanPolarityClues-2012
GermanPolarityClues-Negative-21042012.tsv  GermanPolarityClues-Neutral-21042012.tsv  GermanPolarityClues-Positive-21042012.tsv
GermanPolarityClues-Negative-Lemma-21042012.tsv  GermanPolarityClues-Neutral-Lemma-21042012.tsv  GermanPolarityClues-Positive-Lemma-21042012.tsv
l-bash-4.1$
```

You will use three of the above files in the folder GermanPolarityClues-2012 for this lab.

### 3. Create directories in HDFS

Run the following commands for creating directories:

```
$ hdfs dfs -mkdir project
$ hdfs dfs -mkdir project/tables
$ hdfs dfs -mkdir project/tables/us
$ hdfs dfs -mkdir project/tables/uk
$ hdfs dfs -mkdir project/tables/de
$ hdfs dfs -mkdir project/tables/fr
$ hdfs dfs -mkdir project/tables/dictionary_us
$ hdfs dfs -mkdir project/tables/dictionary_ge
$ hdfs dfs -mkdir project/tables/dictionary_fr
```

You can view the directories created by the following command:

```
$ hdfs dfs -ls project/tables
```

```
-bash-4.1$ hdfs dfs -ls project/tables
Found 9 items
drwxr-xrwx - mmishra2 hdfs 0 2018-11-01 06:08 project/tables/de
drwxr-xrwx - mmishra2 hdfs 0 2018-10-31 06:13 project/tables/dictionary_ge
drwxr-xrwx - mmishra2 hdfs 0 2018-10-30 02:11 project/tables/dictionary_us
drwxr-xrwx - bdcscce_admin hdfs 0 2018-11-02 17:32 project/tables/four
drwxr-xrwx - mmishra2 hdfs 0 2018-11-01 06:07 project/tables/fr
drwxr-xrwx - bdcscce_admin hdfs 0 2018-11-01 19:19 project/tables/three
drwxr-xrwx - bdcscce_admin hdfs 0 2018-11-01 07:41 project/tables/two
drwxr-xrwx - mmishra2 hdfs 0 2018-11-01 06:07 project/tables/uk
drwxr-xrwx - mmishra2 hdfs 0 2018-11-01 06:07 project/tables/us
-bash-4.1$
```

## 4. Put files in HDFS directories

Run the following commands to put dictionaries into respective folders:

```
$ hdfs dfs -put dictionary_us.tsv project/tables/dictionary_us/
$ hdfs dfs -put GermanPolarityClues-2012/GermanPolarityClues-Negative-
21042012.tsv project/tables/dictionary_ge/
$ hdfs dfs -put GermanPolarityClues-2012/GermanPolarityClues-Neutral-21042012.tsv
project/tables/dictionary_ge/
$ hdfs dfs -put GermanPolarityClues-2012/GermanPolarityClues-Positive-
21042012.tsv project/tables/dictionary_ge/
$ hdfs dfs -put FEEL.csv project/tables/dictionary_fr/
```

You can run the following commands to check the files are there:

```
$ hdfs dfs -ls project/tables/dictionary_ge/

$ hdfs dfs -ls project/tables/dictionary_us/

$ hdfs dfs -ls project/tables/dictionary_fr/
```

```
-bash-4.1$ hdfs dfs -ls project/tables/dictionary_ge/
Found 3 items
-rw-r--r--  2 mmishra2 hdfs      956889 2018-11-17 21:28 project/tables/dictionary_ge/GermanPolarityClues-Negative-21042012.tsv
-rw-r--r--  2 mmishra2 hdfs      60317 2018-11-17 21:32 project/tables/dictionary_ge/GermanPolarityClues-Neutral-21042012.tsv
-rw-r--r--  2 mmishra2 hdfs      846736 2018-11-17 21:32 project/tables/dictionary_ge/GermanPolarityClues-Positive-21042012.tsv
-bash-4.1$ hdfs dfs -ls project/tables/dictionary_us/
Found 1 items
-rw-r--rw-  2 mmishra2 hdfs      308921 2018-10-30 02:11 project/tables/dictionary_us/dictionary_us.tsv
```

Similarly run the following commands to place data review files in the corresponding folders:

```
$ hdfs dfs -put review_uk.tsv project/tables/uk/
$ hdfs dfs -put review_us.tsv project/tables/us/
$ hdfs dfs -put review_fr.tsv project/tables/fr/
$ hdfs dfs -put review_de.tsv project/tables/de/
```

Run the following command to provide permission to the files under project folder:

```
hdfs dfs -chmod -R o+w project/
```

## 5. Creating Hive tables to query data

The following Hive statement creates an external table that allows Hive to query data stored in HDFS. External tables preserve the data in the original file format, while allowing Hive to perform queries against the data within the file.

Open another terminal and login into your account using ssh as in Step 1.

Open **beeline** CLI (Command Line Shell Interface) that is equivalent to **hive** CLI environment as follows. **Beeline** is for multiple users' access to Hive Server 2 of a Hadoop cluster. Press enter without any password when it asks for password.

**NOTE:** the following connect url is an example and it should be given by the instructor:

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

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

```
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;zooKeeper
Namespace=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;zooKeeper
Namespace=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.

Now you have to create your database with your username to separate your tables with other users. For example, the user **mmishra2** should run the following:

**NOTE:** you have to use your username.

```
0: jdbc:hive2://cis5200-bdcsce-4.compute-6082> CREATE DATABASE mmishra2;
```

```
No rows affected (0.277 seconds)
```

```
0: jdbc:hive2://cis5200-bdcsce-4.compute-6082> show DATABASES;
```

```
+-----+
| database_name |
+-----+
| default      |
| mmishra2     |
| ngupta8      |
| whu4         |
+-----+
```

4 rows selected (0.232 seconds)

0: jdbc:hive2://cis5200-bdcsce-4.compute-6082> use mmishra2;

No rows affected (0.184 seconds)

In the beeline shell CLI, you need to copy and paste the following HiveQL code to create external tables.

**NOTE: Don't forget to replace **mmishra2** to your account name in the following HQL code.**

```
DROP TABLE IF EXISTS dictionary_us;  
  
CREATE EXTERNAL TABLE if not exists dictionary_us (  
type string,  
length int,  
word string,  
pos string,  
stemmed string,  
polarity string )  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
STORED AS TEXTFILE  
LOCATION '/user/mmishra2/project/tables/dictionary_us';
```

```
DROP TABLE IF EXISTS dictionary_ge;  
  
CREATE EXTERNAL TABLE if not exists dictionary_ge (  
word string,  
word1 string,  
misc string,  
polarity string,  
stemmed string,  
misc1 string)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
STORED AS TEXTFILE  
LOCATION '/user/mmishra2/project/tables/dictionary_ge';
```



```
DROP TABLE IF EXISTS dictionary_fr;

CREATE EXTERNAL TABLE if not exists dictionary_fr (
word string, polarity string,
joy int, fear int, sadness int, anger int, surprise int,
disgust int)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\ '
STORED AS TEXTFILE
LOCATION '/user/mmishra2/project/tables/dictionary_fr';
```

The above commands created dictionary\_us, dictionary\_fr and dictionary\_ge tables which will later be used for sentiment analysis.

You can query the result by running select statement:

```
select * from dictionary_ge limit 10;
select * from dictionary_us limit 10;
select * from dictionary_fr limit 10;
```

Now the product review data table will be created.

```
DROP TABLE IF EXISTS review_fr;

CREATE EXTERNAL TABLE if not exists review_fr (
marketplace string,
customer_id int,
review_id string,
product_id string,
product_parent int,
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 timestamp)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION '/user/mmishra2/project/tables/fr';
```

```
DROP TABLE IF EXISTS review_us;

CREATE EXTERNAL TABLE if not exists review_us (
marketplace string,
customer_id int,
review_id string,
product_id string,
product_parent int,
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 timestamp)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION '/user/mmishra2/project/tables/us';
```

```
DROP TABLE IF EXISTS review_uk;
```

```
CREATE EXTERNAL TABLE if not exists review_uk (  
marketplace string,  
customer_id int,  
review_id string,  
product_id string,  
product_parent int,  
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 timestamp)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
STORED AS TEXTFILE  
LOCATION '/user/mmishra2/project/tables/uk';
```

```

DROP TABLE IF EXISTS review_de;

CREATE EXTERNAL TABLE if not exists review_de (
marketplace string,
customer_id int,
review_id string,
product_id string,
product_parent int,
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 timestamp)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION '/user/mmishra2/project/tables/de';

```

Then, in the beeline shell, you need to check if the tables are shown:

```
show tables;
```

```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> show tables;
```

```

+-----+--+
| tab_name |
+-----+--+
| dictionary_ge |
| dictionary_us |
| review_de    |
| review_fr    |
| review_uk    |
| review_us    |
+-----+--+

```

```
14 rows selected (0.208 seconds)
```

Now you can query the contents of the table:

```
select * from review_de limit 10;
```

You will observe that the column `review_date` is NULL for all the rows. This is because Hive supports reading alternative timestamp formats. To make it work, run the following Hive query:

```
alter table review_de SET SERDEPROPERTIES ("timestamp.formats"="yyyy-MM-dd");  
alter table review_us SET SERDEPROPERTIES ("timestamp.formats"="yyyy-MM-dd");  
alter table review_uk SET SERDEPROPERTIES ("timestamp.formats"="yyyy-MM-dd");  
alter table review_fr SET SERDEPROPERTIES ("timestamp.formats"="yyyy-MM-dd");
```

Now the `review_date` column will show the correct date.

You can see the structure of the table as well

```
describe review_de;
```

```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> describe review_de;  
+-----+-----+-----+  
| col_name | data_type | comment |  
+-----+-----+-----+  
| marketplace | string | |  
| customer_id | int | |  
| review_id | string | |  
| product_id | string | |  
| product_parent | int | |  
| 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 | timestamp | |  
+-----+-----+-----+  
15 rows selected (0.211 seconds)
```

## 6. Creating Hive Queries to Analyze Data

You will create a base table named review which will have all the data from the tables review\_us, review\_uk, review\_de and review\_fr. All the subsequent queries will be based on the new created table.

```
DROP TABLE IF EXISTS review;

CREATE TABLE review AS
select * from review_de where review_id is not null and star_rating is not null
union
select * from review_fr where review_id is not null and star_rating is not null
union
select * from review_uk where review_id is not null and star_rating is not null
union
select * from review_us where review_id is not null and star_rating is not null;
```

The 'Where' clause is used in the above table for cleaning any junk data.

You can query the table using following select statement:

```
select * from review limit 5;
```

### 1. The below query will return the review count group by star rating.

```
select count(review_id) count, star_rating from review
group by star_rating order by star_rating;
```

count	star_rating
554283	1
383986	2
738964	3
1714006	4
6180622	5

Now a table will be created using this query and stored in HDFS for visualization at a later stage.

```
CREATE TABLE IF NOT EXISTS rating ROW FORMAT DELIMITED
FIELDS TERMINATED BY "," STORED AS TEXTFILE
LOCATION "/user/mmishra2/project/tables/one" AS
SELECT COUNT(review_id)count ,star_rating FROM review
GROUP BY star_rating ORDER BY star_rating;
```

Switch on to the first terminal. You can see the directory "one" has been created under project/tables and if you view the directory you can see a file has been placed there.

```
-bash-4.1$ hdfs dfs -ls project/tables/one/
```

Found 1 items

```
-rwxr-xrwx  2 bdcscs_admin hdfs      47 2018-11-18 08:12 project/tables/one/000000_0
```

You can view the contents of the file with the below command:

```
-bash-4.1$ hdfs dfs -cat project/tables/one/000000_0
```

```
554283,1
```

```
383986,2
```

```
738964,3
```

```
1714006,4
```

```
6180622,5
```

**2. The below command will create table based on top ten highest number of reviews given by unique users over ten years.**



```
CREATE TABLE IF NOT EXISTS users ROW FORMAT DELIMITED
FIELDS TERMINATED BY "," STORED AS TEXTFILE
LOCATION "/user/mmishra2/project/tables/two" AS
SELECT COUNT(review_id) count, customer_id FROM review
GROUP BY customer_id ORDER BY count DESC LIMIT 10;
```

This created a table “users” and stored it as a text file under project/tables/two directory.

### **3. The below command will create table based on number of review count grouped by year and month**

```
CREATE TABLE IF NOT EXISTS review_date ROW FORMAT
DELIMITED
FIELDS TERMINATED BY "," STORED AS TEXTFILE
LOCATION "/user/mmishra2/project/tables/three" AS
SELECT COUNT(review_id) count, YEAR(review_date) year,
MONTH(review_date) month FROM review
GROUP BY YEAR(review_date), MONTH(review_date)
ORDER BY year, month;
```

This created a table “review\_date” and stored it as a text file under project/tables/three directory.

### **4. The below command will create table based on review count by product category**

```
CREATE TABLE IF NOT EXISTS product ROW FORMAT DELIMITED
FIELDS TERMINATED BY "," STORED AS TEXTFILE
LOCATION "/user/mmishra2/project/tables/four" AS
SELECT COUNT(review_id) count, product_category FROM
review
```

This created a table “product” and stored it as a text file under project/tables/four directory.

### **5. The below command will create table for top ten popular products based on average rating of 5 and having maximum review count.**

```
CREATE TABLE IF NOT EXISTS popular ROW FORMAT DELIMITED
FIELDS TERMINATED BY "," STORED AS TEXTFILE
LOCATION "/user/mmishra2/project/tables/five" AS
SELECT COUNT(review_id)count, AVG(star_rating) rating,
product_title, product_category, marketplace FROM review
GROUP BY product_title, product_category, marketplace
HAVING AVG(star_rating) = 5 ORDER BY count DESC LIMIT 10;
```

This created a table “popular” and stored it as a text file under project/tables/five directory.

## 6. The below command will create table showing rating of product category “Baby” by country

```
CREATE TABLE IF NOT EXISTS baby ROW FORMAT DELIMITED
FIELDS TERMINATED BY "," STORED AS TEXTFILE
LOCATION "/user/mmishra2/project/tables/seven" AS
SELECT review_id, star_rating, marketplace, review_date
FROM review WHERE product_category = 'Baby';
```

This created a table “baby” and stored it as a text file under project/tables/seven directory.

## 7. Views and tables mentioned below will analyze overall sentiments for countries – US, UK, France and Germany.

```
CREATE VIEW IF NOT EXISTS v1 AS
SELECT marketplace, review_id, review_body FROM review;

CREATE VIEW IF NOT EXISTS v2 AS
SELECT marketplace, review_id, words FROM v1
LATERAL VIEW EXPLODE(SENTENCES(LOWER(review_body))) dummy
as words;

CREATE VIEW IF NOT EXISTS v3 AS
SELECT marketplace, review_id, word FROM v2
LATERAL VIEW EXPLODE(words) dummy as word;
```

You will use US dictionary for countries US and UK. For Germany, you will use German dictionary and for France, you will use French dictionary.

```

CREATE VIEW IF NOT EXISTS v4 AS
SELECT marketplace, review_id, v3.word ,
CASE d_us.polarity
WHEN 'negative' THEN -1
WHEN 'positive' THEN 1
ELSE 0 END AS polarity
FROM v3 LEFT OUTER JOIN dictionary_us d_us on v3.word =
d_us.word
WHERE marketplace = 'US'
UNION
CREATE VIEW IF NOT EXISTS v4 AS
SELECT marketplace, review_id, v3.word,
CASE d_us.polarity
WHEN 'negative' THEN -1
WHEN 'positive' THEN 1
ELSE 0 END AS polarity
FROM v3 LEFT OUTER JOIN dictionary_us d_us on v3.word =
d_us.word
WHERE marketplace = 'UK'
UNION
CREATE VIEW IF NOT EXISTS v4 AS
SELECT marketplace, review_id, v3.word,
CASE d_us.polarity
WHEN 'negative' THEN -1
WHEN 'positive' THEN 1
ELSE 0 END AS polarity
FROM v3 LEFT OUTER JOIN dictionary_fr d_fr on v3.word =
d_us.word
WHERE marketplace = 'FR'
UNION
SELECT marketplace, review_id, v3.word,
CASE d_ge.polarity
WHEN 'negative' THEN -1
WHEN 'positive' THEN 1
ELSE 0 END AS polarity
FROM v3 LEFT OUTER JOIN dictionary_ge d_ge on v3.word =
d_ge.word
WHERE marketplace = 'DE';

```

```

CREATE view IF NOT EXISTS v5 AS
SELECT marketplace, review_id,
CASE
WHEN SUM(polarity) > 0 THEN 'positive'
WHEN SUM(polarity) < 0 THEN 'negative'
ELSE 'neutral' END AS sentiment
FROM v4 GROUP BY marketplace, review_id;

CREATE TABLE IF NOT EXISTS sentiment ROW FORMAT DELIMITED
FIELDS TERMINATED BY "," STORED AS TEXTFILE
LOCATION "/user/mmishra2/project/tables/All_Countries" AS
SELECT * FROM v5;

```

This created a sentiment based table and stored it as a text file under project/tables/All\_Countries directory.

## 8. Ngram sentimental analysis for the least rated product in US

The below query gives the five least rated product in US with review count greater than 50.

```

SELECT product_id, product_title, marketplace,
product_category,
FORMAT_NUMBER(AVG(star_rating),2) AS avg_rating,
COUNT(*) AS num
FROM review_us
GROUP BY product_id, product_title, marketplace,
product_category
HAVING num > 50 ORDER BY avg_rating limit 5;

```

Filtering out the least rated product based on the above query

```

SELECT product_id, product_title, marketplace,
product_category,
FORMAT_NUMBER(AVG(star_rating),2) AS avg_rating,
COUNT(*) AS num
FROM review_us
WHERE product_id = 'B00B5P37IG'
GROUP BY product_id, product_title, marketplace,
product_category;

```

```

+-----+-----+-----+-----+-----+
| product_id | product_title | marketplace | product_category | avg_rating |
+-----+-----+-----+-----+-----+
| B00B5P37IG | Live Tv Streaming (Kindle Tablet Edition) | US | Mobile_Apps | 1.48 |
+-----+-----+-----+-----+-----+
1 row selected (34.636 seconds)

```

Using bigram for text analysis of the least rated product

```

SELECT
EXPLODE (NGRAMS (SENTENCES (LOWER(review_body)), 2,
5))
AS bigrams
FROM review_us
WHERE product_id= 'B00B5P37IG';

```

```

+-----+
|                bigrams                |
+-----+
| {"ngram":["this","app"],"estfrequency":42.0} |
| {"ngram":["i","would"],"estfrequency":22.0}  |
| {"ngram":["waste","of"],"estfrequency":20.0} |
| {"ngram":["i","have"],"estfrequency":17.0}   |
| {"ngram":["to","watch"],"estfrequency":15.0}  |
+-----+
5 rows selected (16.55 seconds)

```

Using trigram for text analysis

```

SELECT
EXPLODE (NGRAMS (SENTENCES (LOWER(review_body)), 3,
5))
AS trigrams
FROM review_us
WHERE product_id= 'B00B5P37IG';

```

```

+-----+
|                trigrams                |
+-----+
| {"ngram":["a","waste","of"],"estfrequency":10.0} |
| {"ngram":["waste","of","money"],"estfrequency":10.0} |
| {"ngram":["i","would","not"],"estfrequency":7.0}   |
| {"ngram":["not","worth","the"],"estfrequency":7.0}  |
| {"ngram":["kindle","fire","hd"],"estfrequency":6.0} |
+-----+
5 rows selected (15.274 seconds)

```

It can be seen that the commonly used word is “waste of”.

Retrieving sentences where the phrase “waste of” is used

```
SELECT review_body
FROM review_us
WHERE product_id= 'B00B5P37IG'
AND review_body LIKE '%waste of %'
LIMIT 3;
```

Based on the above query it can be concluded that the rating and review for the product “Live Tv Streaming (Kindle Tablet Edition)” is bad because of the following:

- Misleading description
- No free choice of streaming
- No support for local channels

On working over the above mentioned issues, the reviews and ratings for the product can be increased.

## 7. Downloading data into your PC

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

1. Switch on to the first terminal connected to the Oracle cloud to download the output file

```
$ ssh mmishra2@ipaddress
mmishra2@ipaddress's password:
```

Run the following command to check if files are present:

```
-bash-4.1$ hdfs dfs -ls project/tables/one/
-bash-4.1$ hdfs dfs -ls project/tables/two/
-bash-4.1$ hdfs dfs -ls project/tables/three/
-bash-4.1$ hdfs dfs -ls project/tables/four/
-bash-4.1$ hdfs dfs -ls project/tables/five/
-bash-4.1$ hdfs dfs -ls project/tables/seven/
-bash-4.1$ hdfs dfs -ls project/tables/All_Countries/
```

```
-bash-4.1$ hdfs dfs -ls project/tables/one/
Found 1 items
-rwxr-xrwx  2 bdcscce_admin hdfs      47 2018-11-18 09:57 project/tables/one/000000_0
```

You will see only one file named 000000\_0 is present in all the above folders except at project/tables/four/, project/tables/seven/ and project/tables/All\_Countries/. The latter locations have multiple and will need a merge.





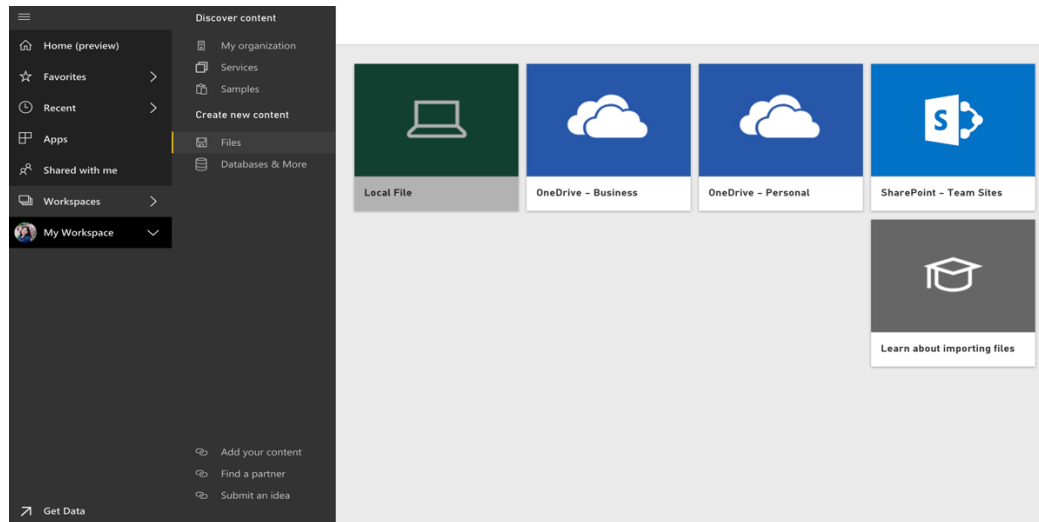
Similarly, do it for others

```
$ scp mmishra2@ipaddress:/home/mmishra2/two two.csv
$ scp mmishra2@ipaddress:/home/mmishra2/three three.csv
$ scp mmishra2@ipaddress:/home/mmishra2/four four.csv
$ scp mmishra2@ipaddress:/home/mmishra2/five five.csv
$ scp mmishra2@ipaddress:/home/mmishra2/seven seven.csv
$ scp mmishra2@ipaddress:/home/mmishra2/All_Countries
All_Countries.csv
```

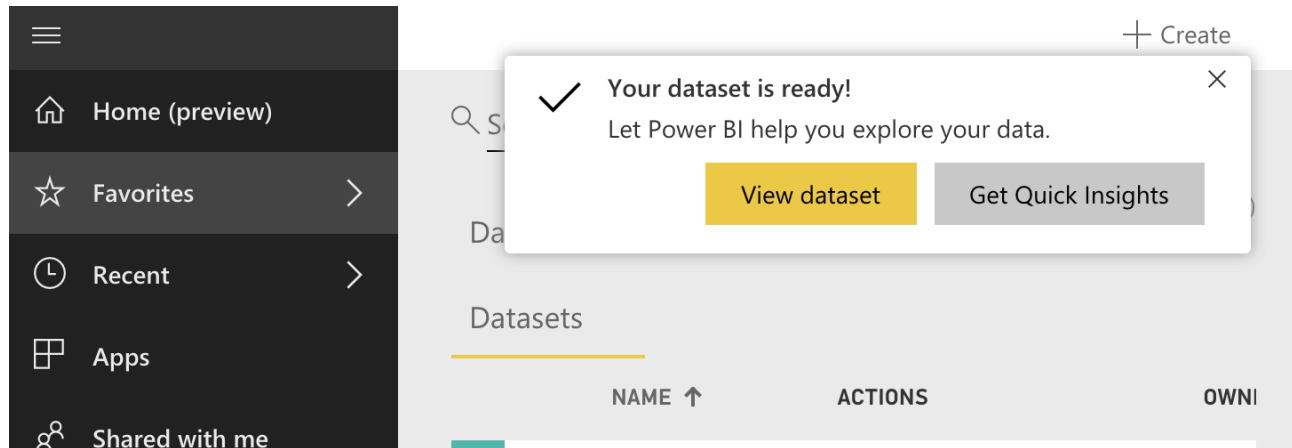
## 8. Visualizing data

### 1. Review count group by star rating

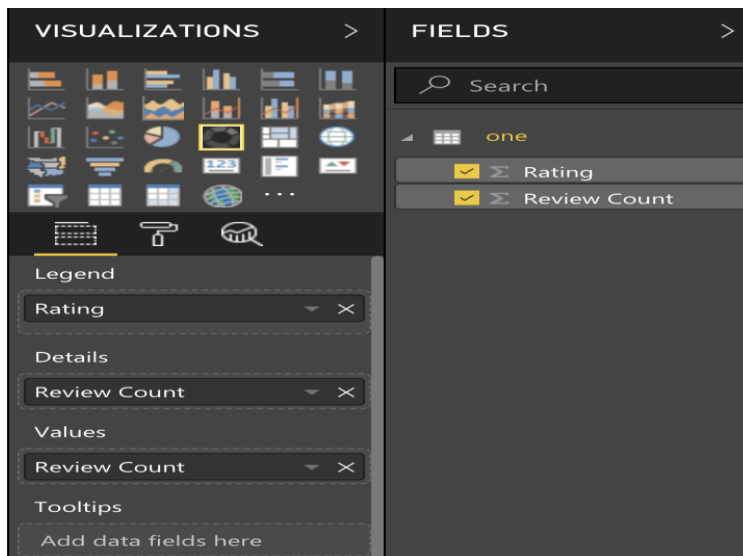
- a. Open the one.csv file with Microsoft Excel and insert column heading as Review Count and Rating. Save the document.
- b. Open a web browser and go to and sign in with your school account at:  
<https://app.powerbi.com>
- c. Click on Local File and select the one.csv



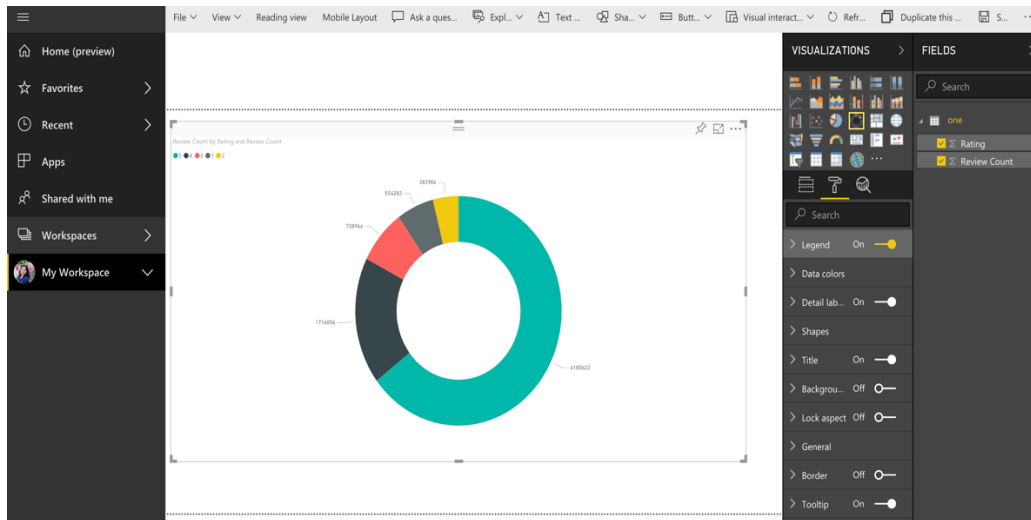
- d. Click on View dataset once the file is uploaded



- e. Click on Donut chart under VISUALIZATIONS.  
Drag Rating under Legend. Drag Review Count under Details and values

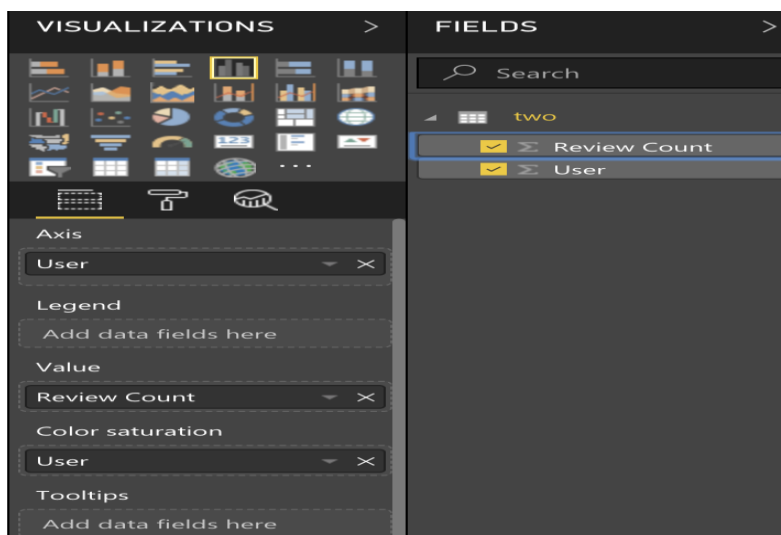


- f. Click on the format under VISUALIZATIONS and switch on the Legend  
You can customize colors and other features using format



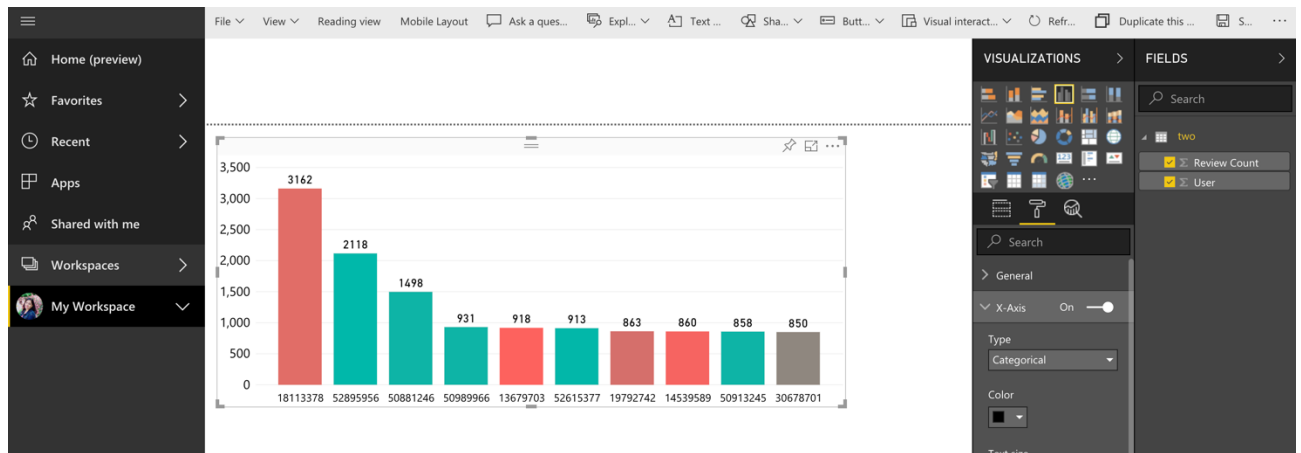
## 2. Number of reviews given by unique users

- Open the two.csv file with Microsoft Excel and insert column heading as Review Count and User. Save the document.
- Open a web browser and go to and sign in with your school account at: <https://app.powerbi.com>
- Click on Local File and select the two.csv
- Click on View dataset once the file is uploaded
- Click on Clustered column chart under VISUALIZATIONS. Drag User under Axis and Color saturation. Drag Review Count under Value



- f. Click on the format under VISUALIZATIONS and select type as “categorical” under X-Axis.

You can customize colors and other features using format



### 3. Review count by year and month

- a. Open the three.csv file with Microsoft Excel and insert column heading as Count, Year and Month. Save the document.
- b. Open a web browser and go to and sign in with your school account at: <https://app.powerbi.com>
- c. Click on Local File and select the three.csv
- d. Click on View dataset once the file is uploaded
- e. Click on Clustered bar chart under VISUALIZATIONS.  
Drag Year and month under Axis.  
Drag month under legend.  
Drag Count under Value
- f. You will display values from 2010 to 2014 and so under filter section for Year select as per the below screenshot and click apply filter

FILTERS

Visual level filters

Count

is (All)

Month

is (All)

Year

is greater than 2009 a...

Filter type

Advanced filtering

Show items when the value:

is greater than

2009

☒ And

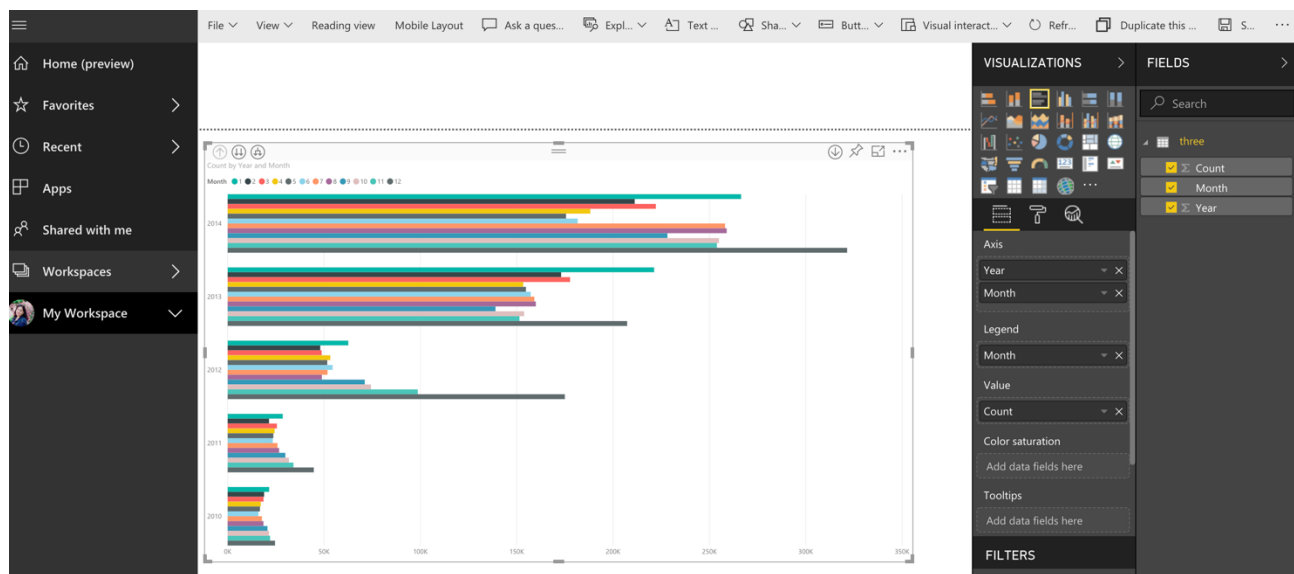
☐ Or

is less than

2015

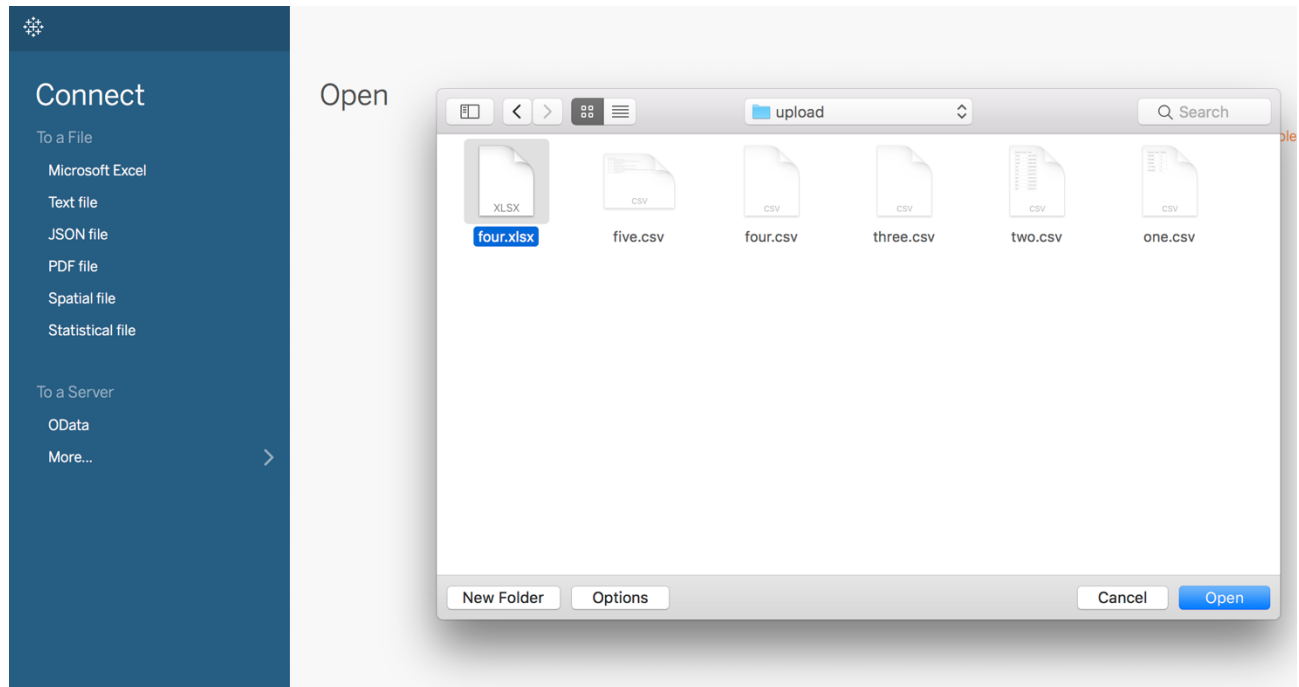
Apply filter

The visualization will be like as below:

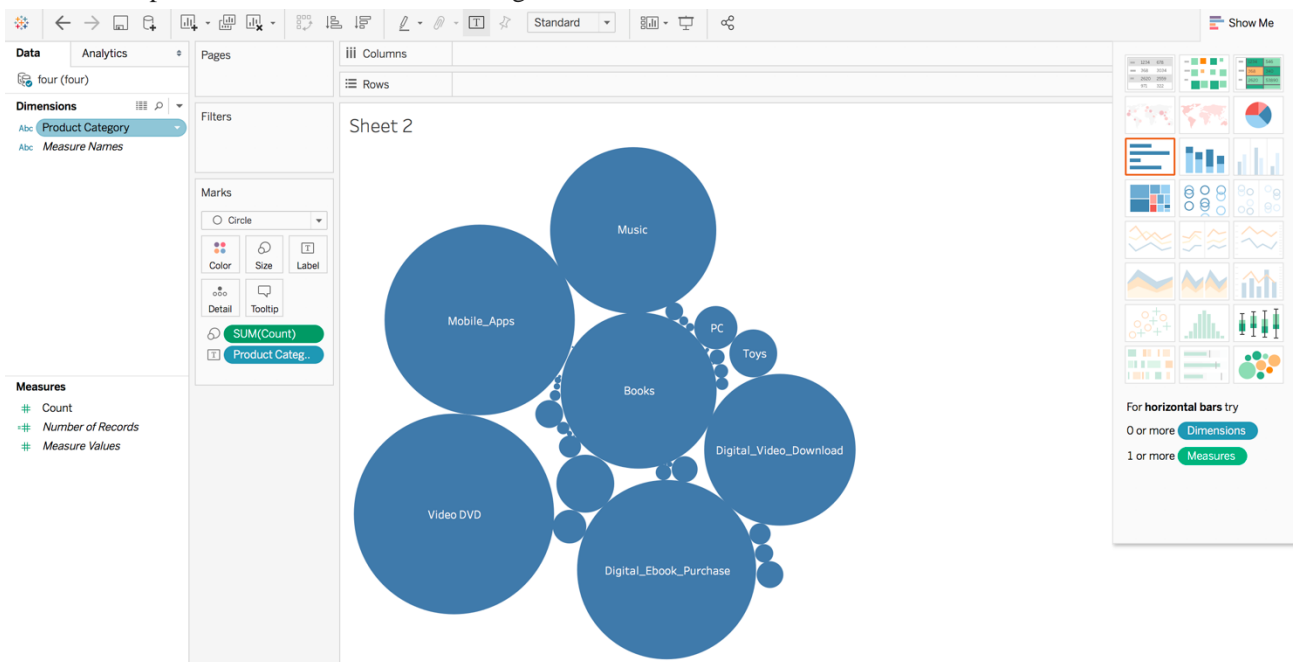


## 4. Review count by product category

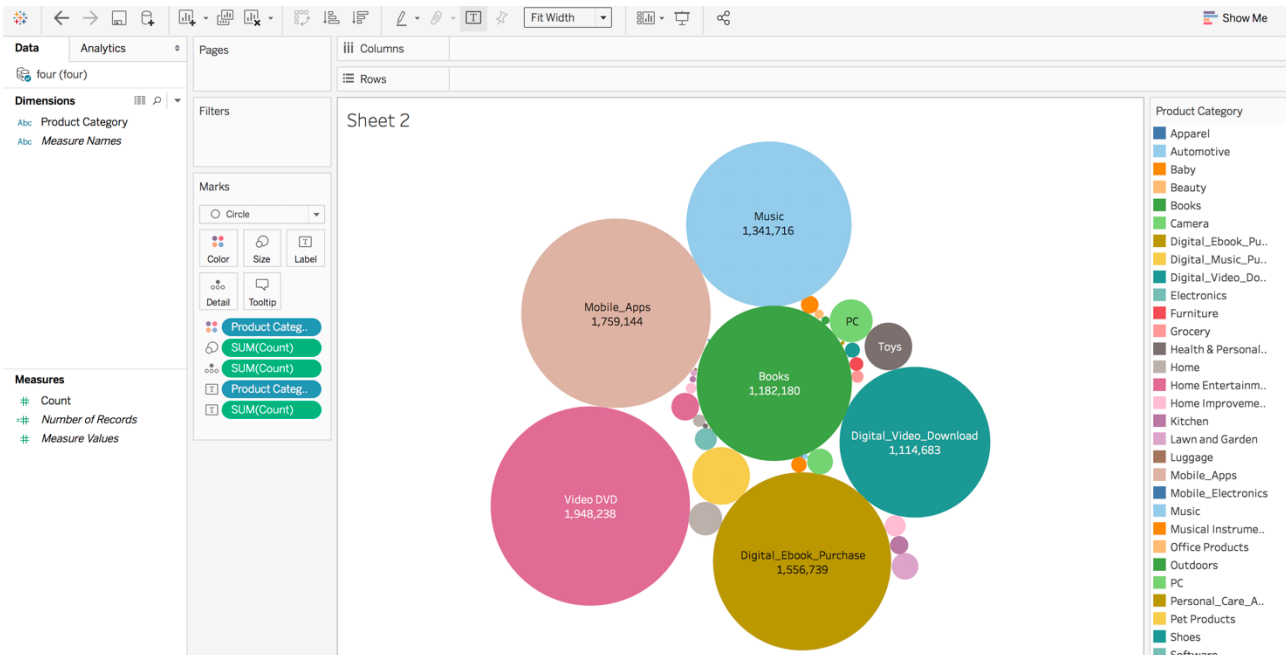
- Open the four.csv file with Microsoft Excel and insert column heading as Count and Product Category. Save the document in xlsx format.
- Open the tableau in your machine
- Click on Microsoft Excel under Connect and select the four.xlsx file



- d. Click on the New worksheet at the bottom
  - e. Drag Product Category under Rows and Count under columns
- Click on the packed bubbles at the extreme right

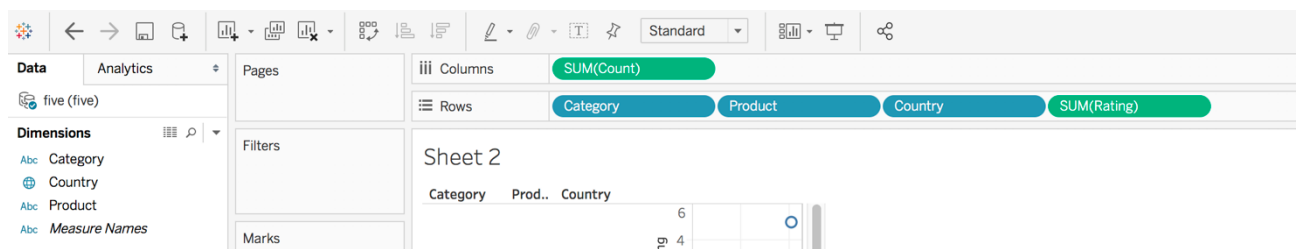


- f. You can drag Product Category from Dimensions and Count from Measures on the extreme right to Marks which will customize your visualization



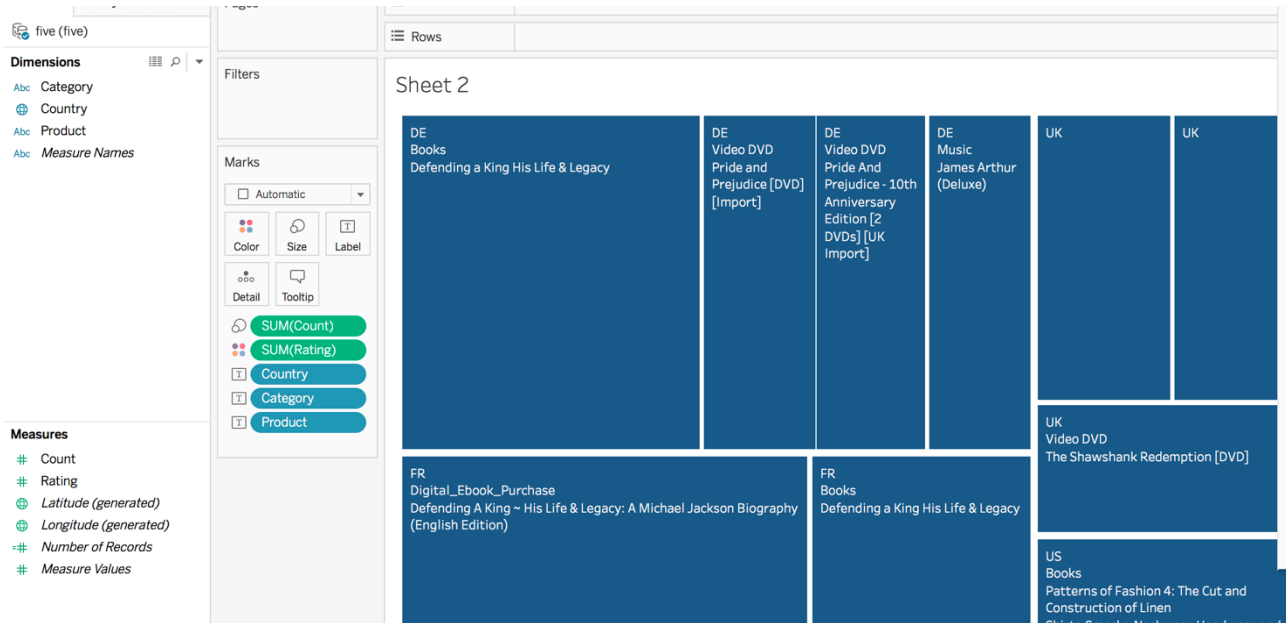
## 5. Popular product based on average rating and review Count

- Open the five.csv file with Microsoft Excel and insert column heading as Count, Rating, Product, Category and Country. Save the document in xlsx format.
- Open the tableau in your machine
- Click on Microsoft Excel under Connect and select the five.xlsx file
- Click on the New worksheet at the bottom
- Drag Category, Product, Country and Rating under Rows and Count under columns



Click on the treemaps at the right

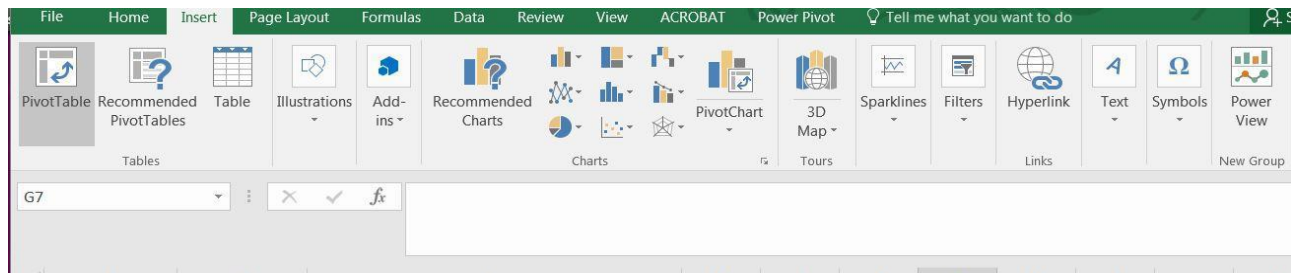




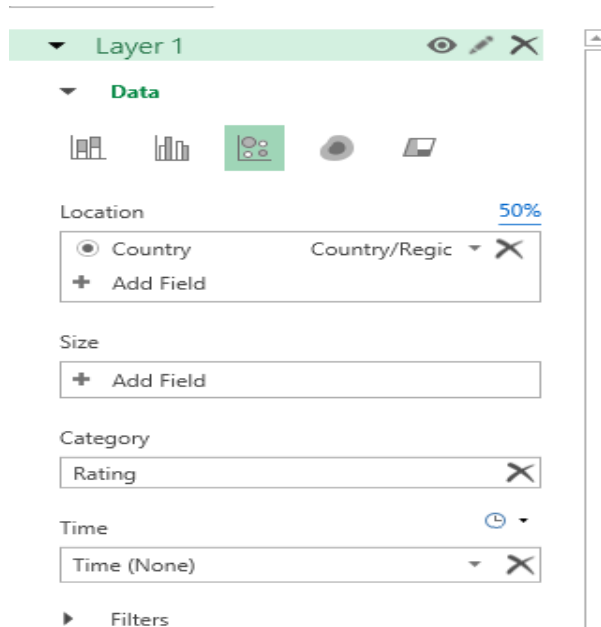
- f. You can drag count measures more than once to arrange it by colors and sizes. You can customize as per your wish

## 6. Rating based geospatial representation of product category Baby

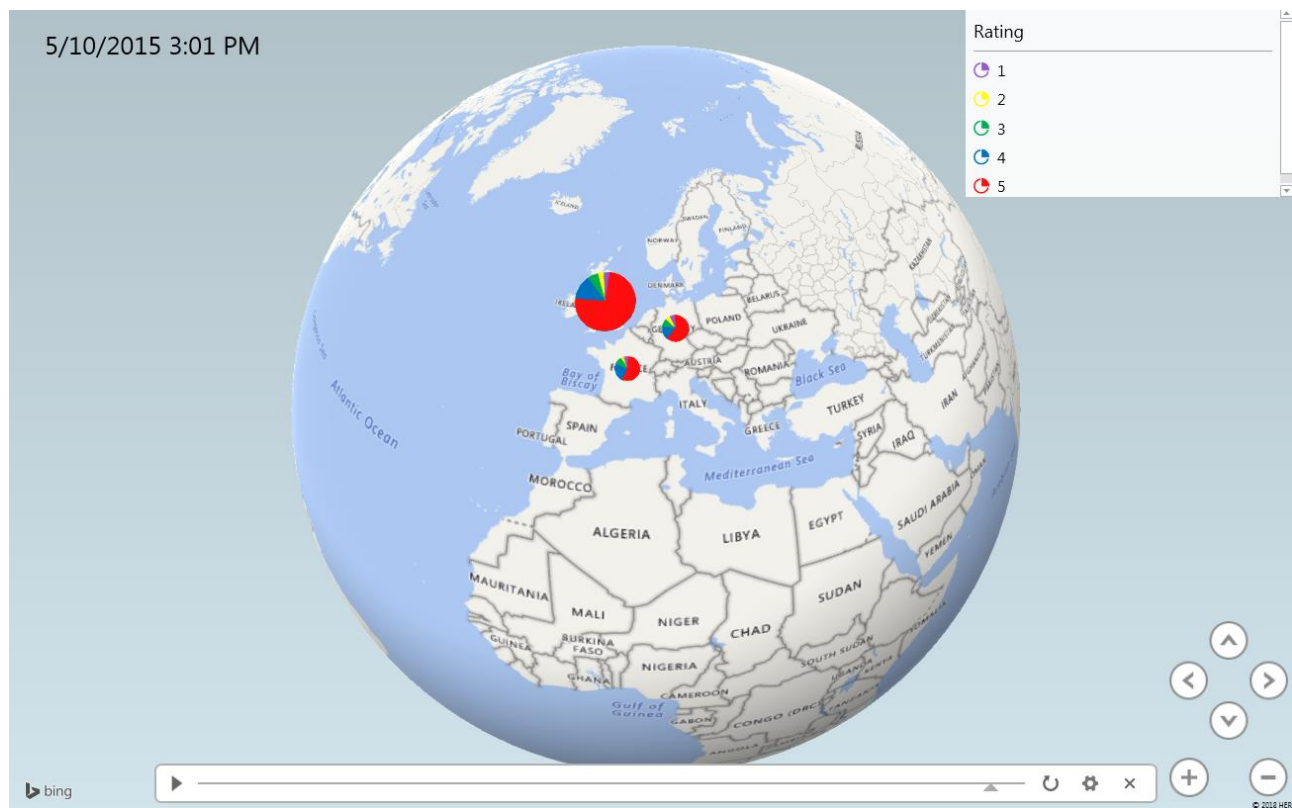
- a. Open the seven.csv file with Microsoft Excel and insert column heading as Review Id, Rating, Country and Time. Save in.xlsx.
- b. Open seven.xlsx in MS-excel. Go to Insert tab and click on 3D Map.



- c. You will see the 3D map.  
**NOTE:** If you don't see the layer frame in the right side, you may select all data manually before opening 3D map
- d. Add fields as shown in the picture below

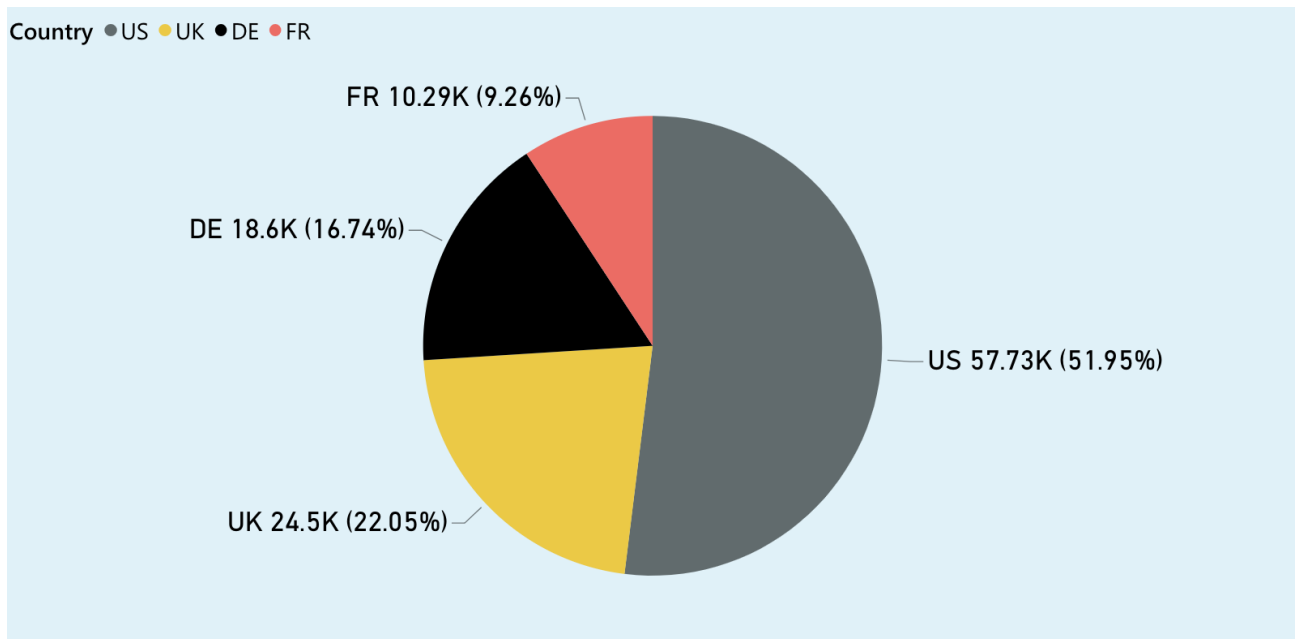


- e. You will get a view like below. You can click on **play** button to observe how much reviews have been generating



## 7. Overall sentiment count by country

- Open the All\_Countries.csv file with Microsoft Excel and insert column heading as Country, Review, Sentiment
- Open a web browser and go to and sign in with your school account at: <https://app.powerbi.com>
- Click on Local File and select the All\_Countries.csv
- Click on View dataset once the file is uploaded
- Click on Pie chart under VISUALIZATIONS.  
Drag Country under Axis.  
Drag sentiment under legend.  
Drag Count of sentiment under Value
- You can customize the colors as per your wish



## 8. Overall sentiment analysis – positive, negative, neutral by country

- Use the same All\_Countries.csv file as above
- Click on Stacked column chart under VISUALIZATIONS in Power BI.  
Drag Country under Axis.  
Drag sentiment under legend.  
Drag Count of sentiment under Value
- You can customize the colors as per your wish

Sentiment ● negative ● neutral ● positive

60K

50K

40K

30K

20K

10K

0K

41K

12K

5K

US

18K

4K

2K

UK

14K

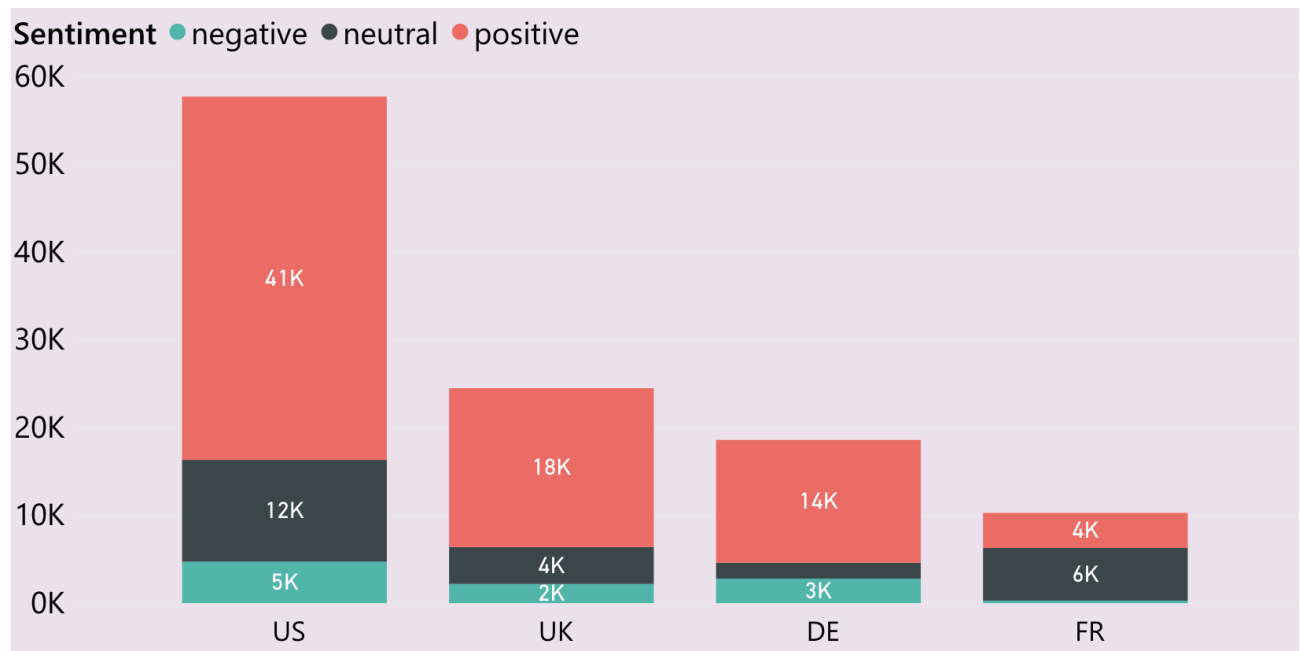
3K

DE

4K

6K

FR



# Summary

In this tutorial you learned how Oracle Cloud Big Data can be used to analyze different patterns of raw data using Apache Hive. You went through a flow to understand how the raw data is first uploaded to HDFS, and then loaded to Hive tables for performing queries. And, you learned how to import the results of Hive queries into Microsoft Excel, tableau and powerbi. Finally, you learned how to create visualizations using tableau, powerbi and 3D Map chart in MS-excel.

# References

1. <https://github.com/monika2403/mmishra2>
2. [https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon\\_reviews\\_multilingual\\_US\\_v1\\_00.tsv.gz](https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_US_v1_00.tsv.gz)
3. [https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon\\_reviews\\_multilingual\\_DE\\_v1\\_00.tsv.gz](https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_DE_v1_00.tsv.gz)
4. [https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon\\_reviews\\_multilingual\\_UK\\_v1\\_00.tsv.gz](https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_UK_v1_00.tsv.gz)
5. [https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon\\_reviews\\_multilingual\\_FR\\_v1\\_00.tsv.gz](https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_FR_v1_00.tsv.gz)
6. <https://s3.amazonaws.com/hipicdatasets/dictionary.tsv>
7. <http://www.ulliwaltinger.de/sentiment/GermanPolarityClues-2012.zip>
8. [https://www.dropbox.com/home/cis4560\\_5200Fall2018/labs](https://www.dropbox.com/home/cis4560_5200Fall2018/labs)
9. <https://advance.lirmm.fr/FEEL.csv>