Project Part 1
SHRESHTHA JHA

Data Cleaning

In Amazon Review dataset there are many records with multiple reviews by the same customer for the same product. This may cause ambiguity in analysis and accuracy may be less. Hence it is better to filter such records from the dataset. Also previous years data may be less effective due to changing trends. So for analysis, I have included the data after 2005 and the data with product categories as Wireless, Automotive, Music, Digital_Music_Purchase, Sports, Toys, Digital_Video_Games,Video_Games have been included in my analysis.

Sql queries to create hive tables.

```
create database amazon_review;

drop table amazon_review.amazon_reviews_parquet;


CREATE EXTERNAL TABLE amazon_review.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` DATE,
`year` int)
PARTITIONED BY (
`product_category` string)
--ROW FORMAT DELIMITED
```

```
--STORED AS PARQUET
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
'hdfs:///hive/amazon-reviews-pds/parquet/'
TBLPROPERTIES (
'transient_lastDdlTime'='1583454851');

Msck repair table amazon_review.amazon_reviews_parquet;
```

Creating temp view to filter data including only required product categories.

```
create view temp
as
select * from amazon_review.amazon_reviews_parquet where review_id in (select review_id from
(select customer_id, product_id,review_id,count(*)
from amazon_review.amazon_reviews_parquet
group by customer_id, product_id,review_id
having count(*)=1) as t) and product_category in
('Wireless','Automotive','Music','Digital_Music_Purchase','Sports','Toys','Digital_Video_Games','Video_G
ames');
```

Creating table to filter reviews that are reviewed multiple times by same customer for same product.

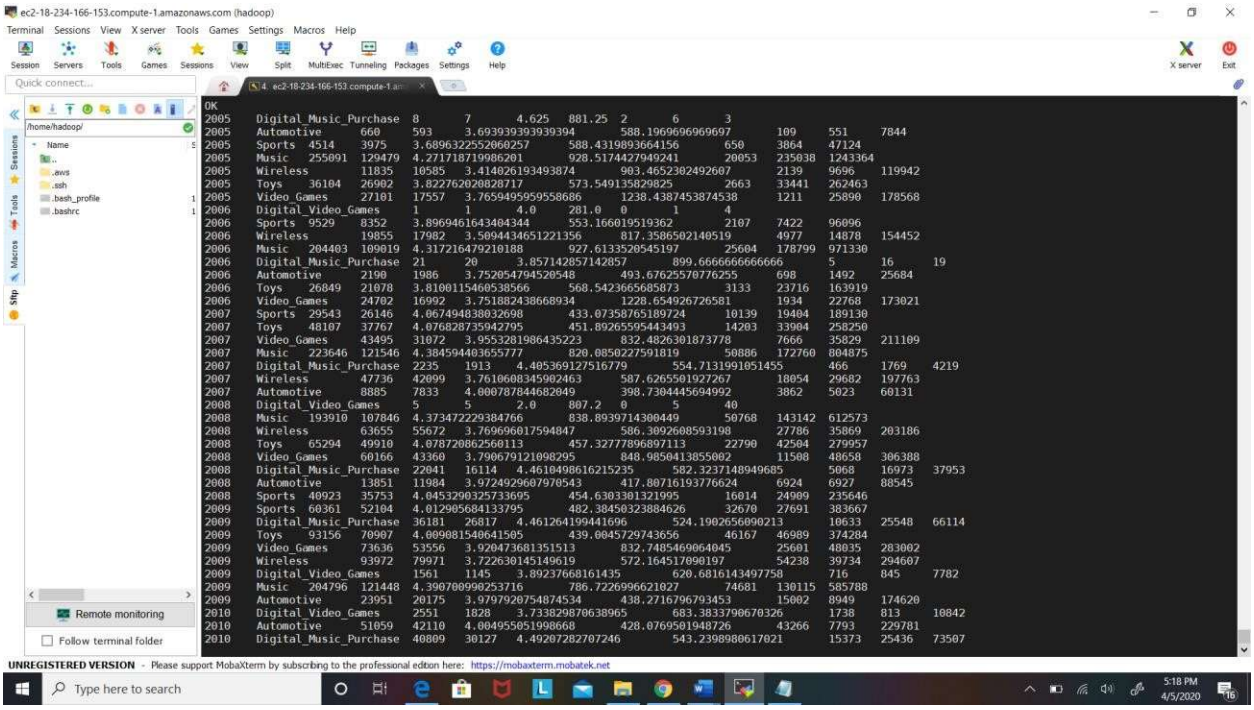
```
create table amazon_review.filtered_reviews
AS
select t.* from(
select *,row_number() over(partition by customer_id,product_id) as row1 from temp)t where row1=1;
```

Exploratory Analysis.

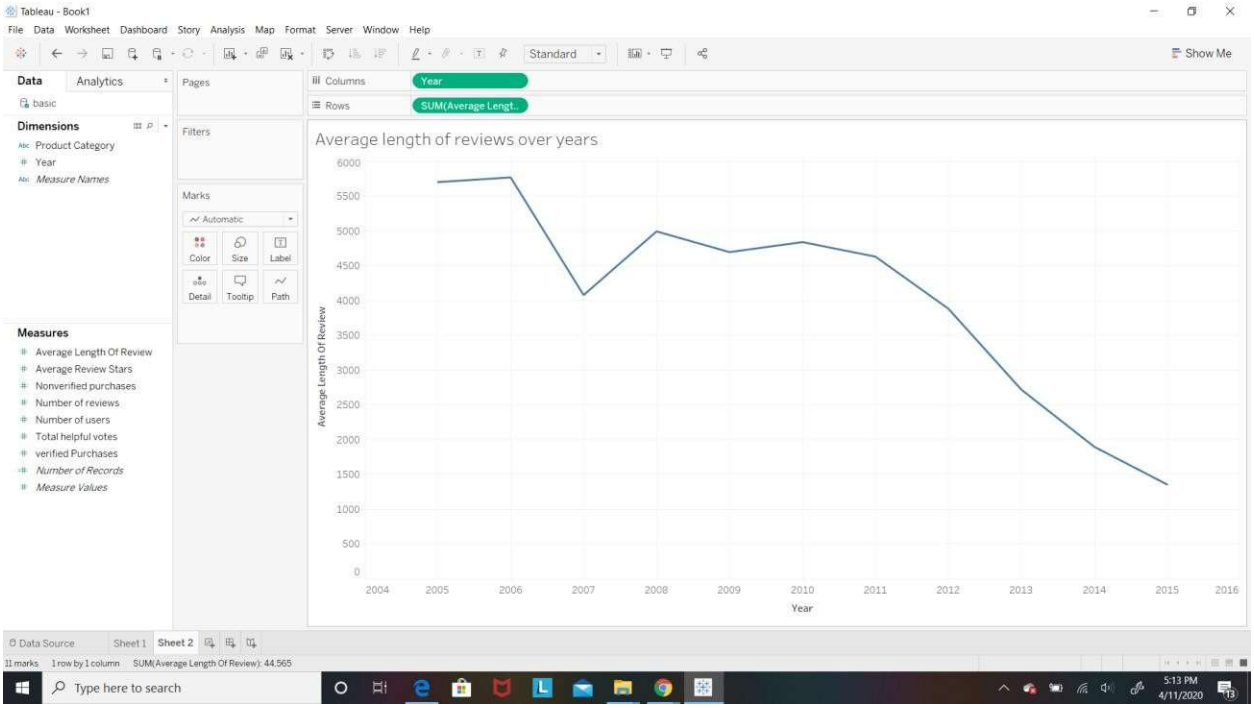
1. **Explore the dataset and provide basic exploratory analysis over time and per product category**

Query-

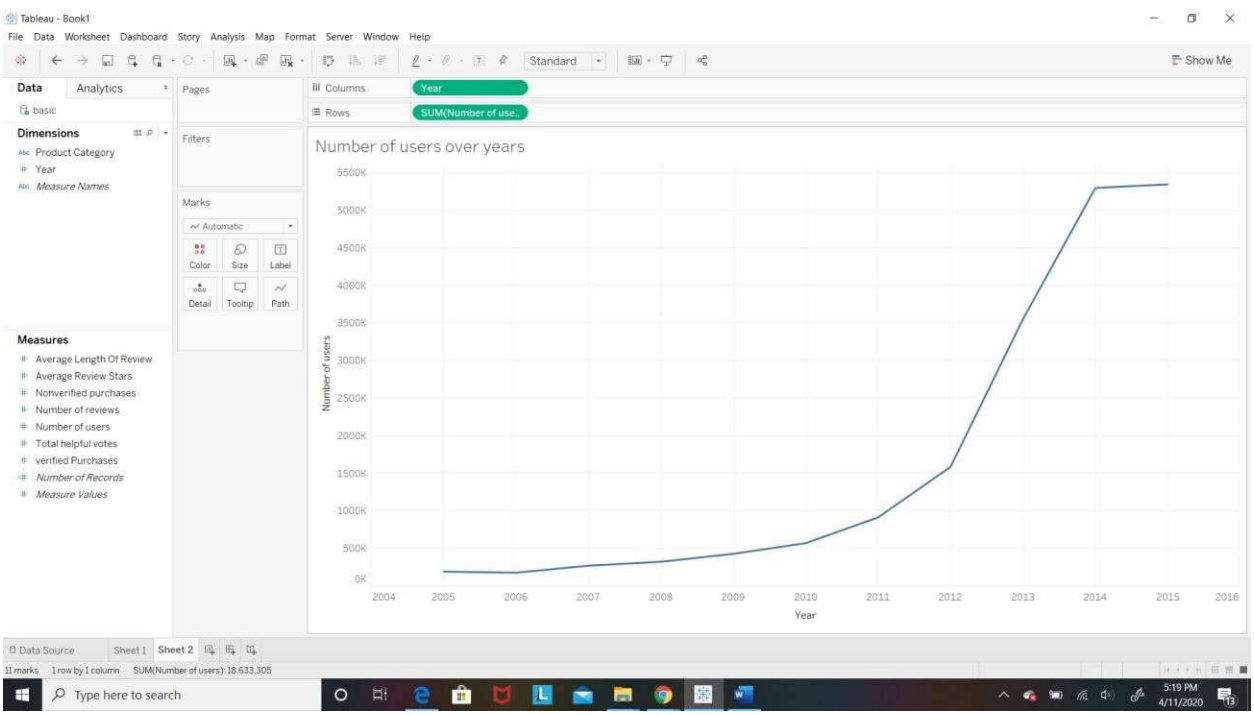
Select year,product_category,count(review_id) as Number_of_reviews,count(Distinct(customer_id)) as Number_of_users,avg(star_rating) as average_review_stars,avg(length(review_body)) as average_length_of_review,
sum(case when verified_purchase='Y' then 1 else 0 end) as verified_Purchases,
sum(case when verified_purchase='N' then 1 else 0 end) as Nonverified_purchases,sum(helpful_votes) as Total_helpful_votes from amazon_review.filtered_reviews where year>=2005 group by year,product_category order by year;



Visualizations



From this visualization we can infer that average length of reviews has decreased over the years.



We can see that Number of users have increased exponentially over the years.

2. Provide detailed analysis of Music/Digital_Music_Purchase and Digital_Video_Games/Video_Games over time.

i Do you see correlation (maybe negative) between the categories over time?

Correlation for Music/Digital_Music_Purchase

Query-

select corr(Music,Digital_Music_Purchase) from (
Select year,sum(case when product_category='Music' then 1 else 0 end) as Music,
sum(case when product_category='Digital_Music_Purchase' then 1 else 0 end) as Digital_Music_Purchase from amazon_review.filtered_reviews where year>=2005 group by year order by year)r;

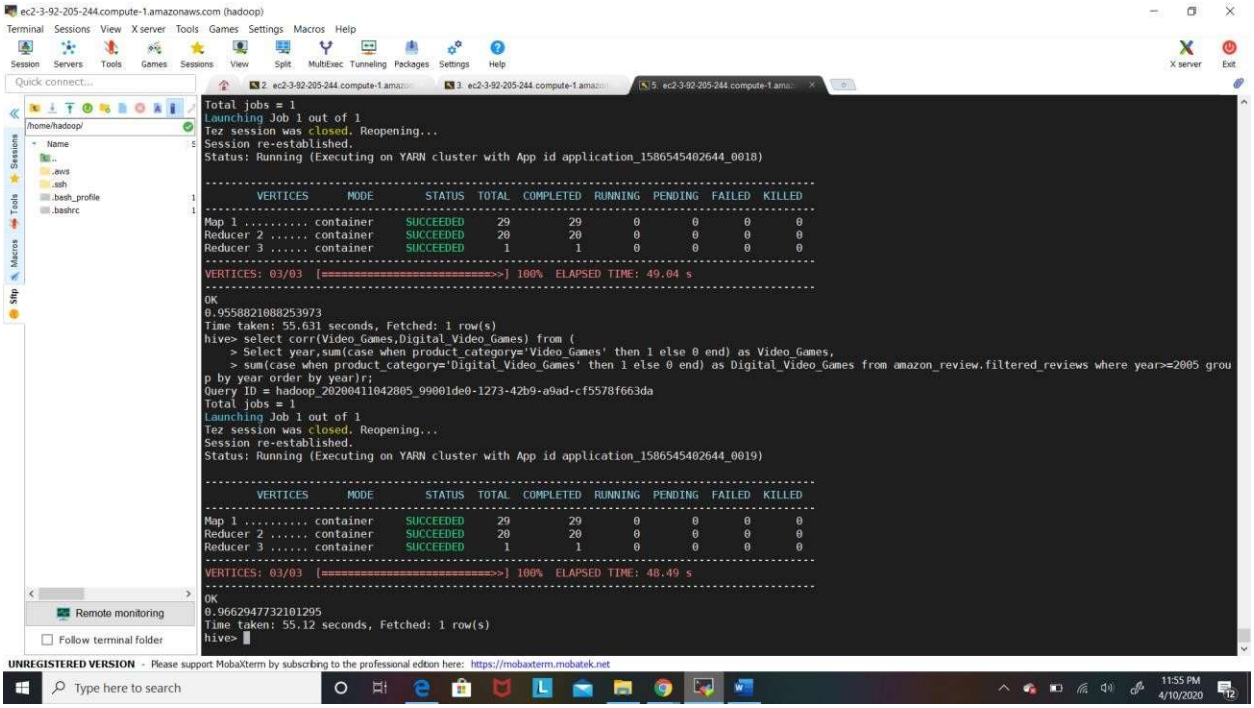
```
hive> select corr(Music,Digital_Music_Purchase) from (
    > Select year,sum(case when product_category='Music' then 1 else 0 end) as Music,
    > sum(case when product_category='Digital_Music_Purchase' then 1 else 0 end) as Digital_Music_Purchase from amazon_review.filtered_reviews where year>=200
5 group by year order by year)r;
Query ID = hadoop_20200411032529_4ef5b5fc-deff-4f60-8c4d-499d977cccdd
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1586545402644_0018)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     29         29        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     20         20        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 49.04 s
--------------------------------------------------------------------------------
OK
0.9558821088253973
Time taken: 55.631 seconds, Fetched: 1 row(s)
hive>
```

Correlation between Digital_Video_Games/Video_Games based on count of reviews.

Query-

select corr(Video_Games,Digital_Video_Games) from (
Select year,sum(case when product_category='Video_Games' then 1 else 0 end) as Video_Games,
sum(case when product_category='Digital_Video_Games' then 1 else 0 end) as Digital_Video_Games from amazon_review.filtered_reviews where year>=2005 group by year order by year)r;

Output-

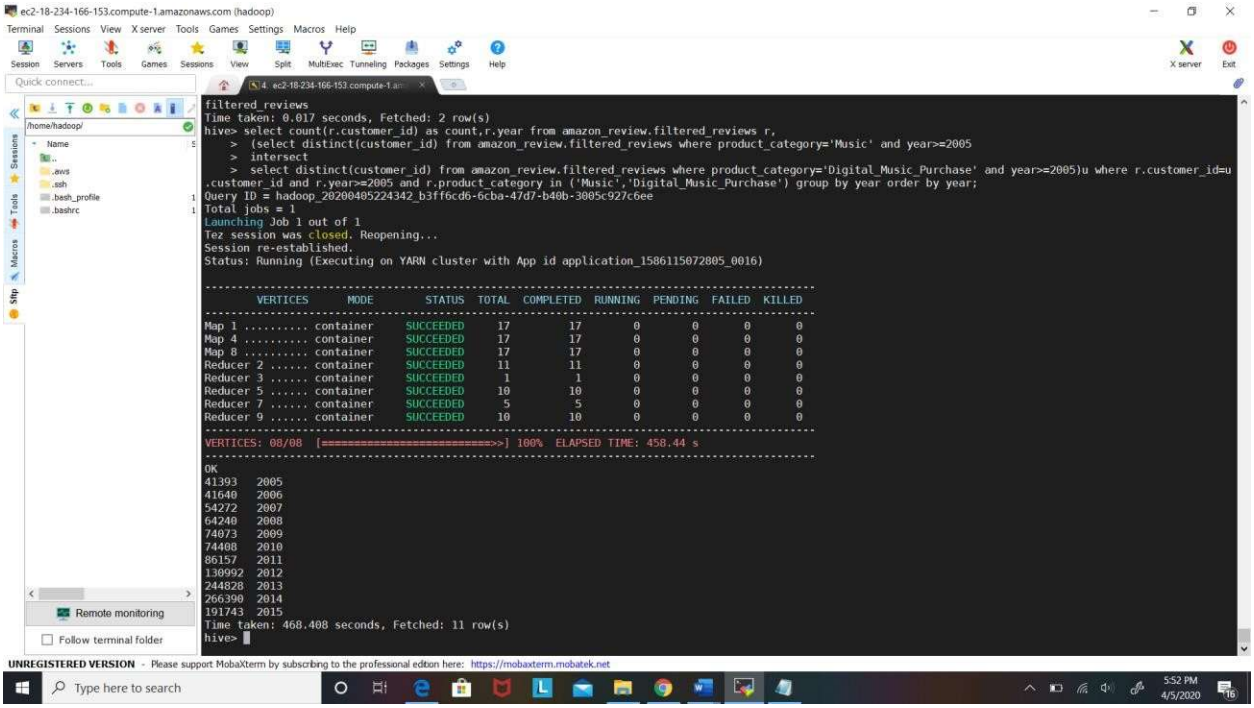


ii Are there same users reviewing in both categories?

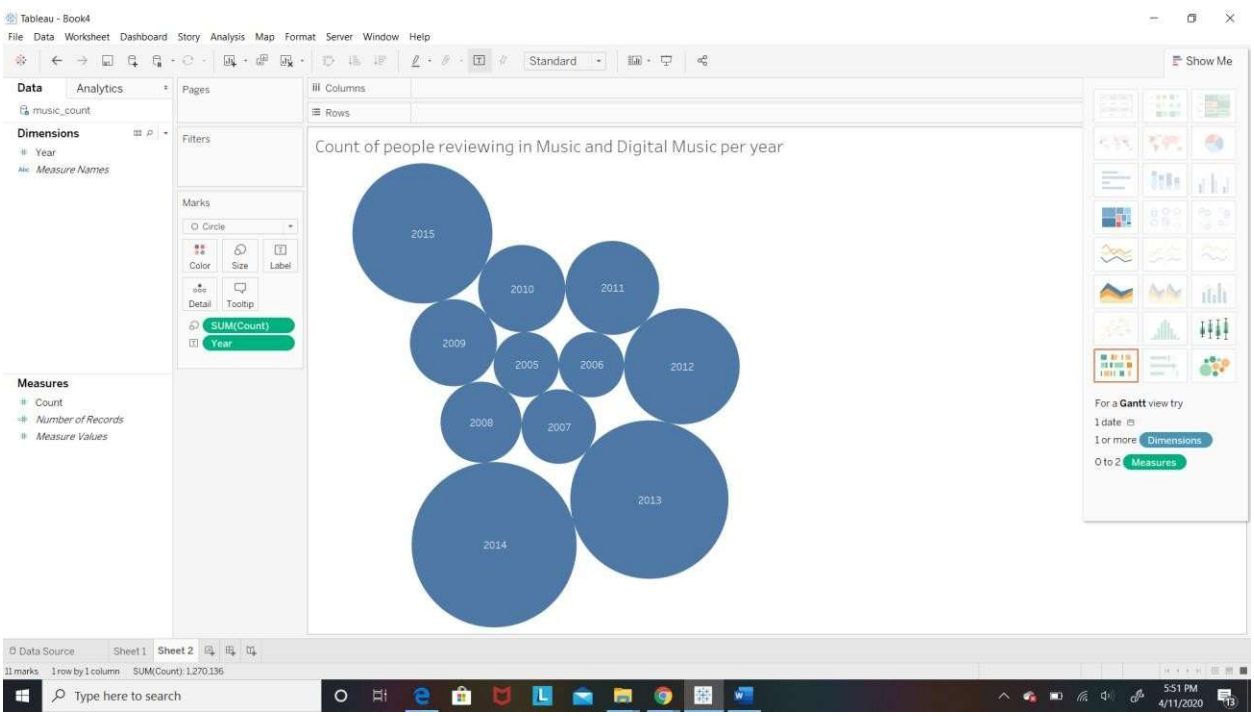
Finding count of users that have given reviews in Music and Digital_Music_Purchase category for each year.

Query-

select count(r.customer_id) as count,r.year from amazon_review.filtered_reviews r,
(select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Music' and year>=2005
intersect
select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Digital_Music_Purchase' and year>=2005)u where r.customer_id=u.customer_id and r.year>=2005 and r.product_category in ('Music','Digital_Music_Purchase') group by year order by year;



Visualization



From this we can infer that 2014 was the year when there were maximum people who had reviewed in both these categories.

Total count of people that have reviewed in both these categories(Music/Digital_Music_Purchase)

Query-

select count(r.customer_id) as count from amazon_review.filtered_reviews r,
(select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Music' and year>=2005
intersect
select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Digital_Music_Purchase' and year>=2005)u where r.customer_id=u.customer_id and r.year>=2005 and r.product_category in ('Music','Digital_Music_Purchase');

```
Time taken: 468.408 seconds, Fetched: 11 row(s)
hive> select count(r.customer_id) as count from amazon_review.filtered_reviews r,
    >  (select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Music' and year>=2005
    >  intersect
    >  select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Digital_Music_Purchase' and year>=2005)u where r.customer_id=u
.customer_id and r.year>=2005 and r.product_category in ('Music','Digital_Music_Purchase');
Query ID = hadoop_20200405225958_8487b00f-f83e-4a7a-ad5e-34887ccb5c47
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1586115072805_0017)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     17         17        0        0       0       0
Map 3 .......... container     SUCCEEDED     17         17        0        0       0       0
Map 7 .......... container     SUCCEEDED     17         17        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 4 ...... container     SUCCEEDED     10         10        0        0       0       0
Reducer 6 ...... container     SUCCEEDED      5          5        0        0       0       0
Reducer 8 ...... container     SUCCEEDED     10         10        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 07/07  [==========================>>] 100%  ELAPSED TIME: 449.51 s
--------------------------------------------------------------------------------
OK
1270136
Time taken: 460.76 seconds, Fetched: 1 row(s)
```

Finding count of users that have given reviews for Digital_Video_Games and Video_Games category over years.

Query-
select count(r.customer_id) as count,r.year from amazon_review.filtered_reviews r,
(select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Digital_Video_Games' and year>=2005
intersect
select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Video_Games' and year>=2005)u where r.customer_id=u.customer_id and r.year>=2005 and r.product_category in ('Video_Games','Digital_Video_Games') group by year order by year;

```
Time taken: 302.521 seconds, Fetched: 4 row(s)
hive> select count(r.customer_id) as count,r.year from amazon_review.filtered_reviews r,
    >  (select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Digital_Video_Games' and year>=2005
    >  intersect
    >  select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Video_Games' and year>=2005)u where r.customer_id=u.customer_i
d and r.year>=2005 and r.product_category in ('Video_Games','Digital_Video_Games') group by year order by year;
Query ID = hadoop_20200406024225_05275c35-7753-42d4-9b9f-106c13c24d0d
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1586115072805_0023)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     17         17        0        0       0       0
Map 4 .......... container     SUCCEEDED     17         17        0        0       0       0
Map 8 .......... container     SUCCEEDED     17         17        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     11         11        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 5 ...... container     SUCCEEDED     10         10        0        0       0       0
Reducer 7 ...... container     SUCCEEDED      5          5        0        0       0       0
Reducer 9 ...... container     SUCCEEDED     10         10        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 08/08  [==========================>>] 100%  ELAPSED TIME: 446.12 s
--------------------------------------------------------------------------------
OK
1082    2005
1484    2006
2158    2007
3883    2008
5842    2009
8063    2010
13001   2011
18912   2012
38267   2013
44350   2014
29991   2015
Time taken: 455.11 seconds, Fetched: 11 row(s)
```

Total Count of people that have reviewed in both these categories.(Digital_Video_Games and Video_Games category)

Query-

select count(r.customer_id) as count from amazon_review.filtered_reviews r,
(select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Digital_Video_Games' and year>=2005
 intersect
 select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Video_Games' and year>=2005)u where r.customer_id=u.customer_id and r.year>=2005 and r.product_category in ('Video_Games','Digital_Video_Games');

```
Time taken: 455.11 seconds, Fetched: 11 row(s)
hive>  select count(r.customer_id) as count from amazon_review.filtered_reviews r,
    >  (select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Digital_Video_Games' and year>=2005
    >  intersect
    >  select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Video_Games' and year>=2005)u where r.customer_id=u.customer_i
d and r.year>=2005 and r.product_category in ('Video_Games','Digital_Video_Games');
Query ID = hadoop_20200406025405_c4532571-e93e-437a-962d-860314b39d0c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586115072805_0023)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     17         17        0        0       0       0
Map 3 .......... container     SUCCEEDED     17         17        0        0       0       0
Map 7 .......... container     SUCCEEDED     17         17        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 4 ...... container     SUCCEEDED     10         10        0        0       0       0
Reducer 6 ...... container     SUCCEEDED      5          5        0        0       0       0
Reducer 8 ...... container     SUCCEEDED     10         10        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 07/07  [==========================>>] 100%  ELAPSED TIME: 439.06 s
--------------------------------------------------------------------------------
OK
167033
Time taken: 439.834 seconds, Fetched: 1 row(s)
hive>
```

iii Can you identify similar items in both categories? Do they get same rating?

For Music/Digital_Music_Purchase

Creating view with product_category music.

create view music as
select product_id,round(avg(star_rating),2) as Average_rating_by_customer_for_Music_products from filtered_reviews where product_category='Music' and year>=2005 group by product_id;

Creating view with product_category digital_Music_Purchase .

create view Digital_Music_Purchase as
select product_id,round(avg(star_rating),2) as Average_rating_by_customer_for_Digital_Music_purchase_products from filtered_reviews where product_category='Digital_Music_Purchase' and year>=2005 group by product_id;

To find similar items in both categories and their average rating in respective categories using inner join on product ids from both views generated above.

Query-

select r.product_id,Average_rating_by_customer_for_Music_products,Average_rating_by_customer_for_Digital_Music_purchase_products from music r inner join Digital_Music_Purchase u on u.product_id=r.product_id;

```
Time taken: 0.151 seconds
hive>   select r.product_id,Average_rating_by_customer_for_Music_products,Average_rating_by_customer_for_Digital_Music_purchase_products from music r inner jo
in Digital_Music_Purchase u on u.product_id=r.product_id;
Query ID = hadoop_20200406014302_36f1ea14-9f4f-4d83-889a-0024fd19c98a
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1586115072805_0021)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     17         17        0        0       0       0
Map 4 .......... container     SUCCEEDED     17         17        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     10         10        0        0       0       0
Reducer 3 ...... container     SUCCEEDED     10         10        0        0       0       0
Reducer 5 ...... container     SUCCEEDED     10         10        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 308.69 s
--------------------------------------------------------------------------------
OK
B0019M1ZJS      3.0     5.0
Time taken: 317.15 seconds, Fetched: 1 row(s)
hive>
```

We can see that this product is common in both product categories but it does not get same average ratings.

For Digital_Video_Games/Video_Games category

Creating view for Digital Video_games category.

create view Digital_Video_Games as
select product_id,round(avg(star_rating),2) as Average_rating_by_customer_for_Digital_Video_Games from filtered_reviews where product_category='Digital_Video_Games' and year>=2005 group by product_id;

Creating view for Video_games Category

create view Video_Games as
select product_id,round(avg(star_rating),2) as Average_rating_by_customer_for_Video_Games from filtered_reviews where product_category='Video_Games' and year>=2005 group by product_id;

To find similar items in both categories and their average rating in respective categories using inner join on product ids from both views generated above.

Query-

select r.product_id,Average_rating_by_customer_for_Digital_Video_Games,Average_rating_by_customer_for_Video_Games from Video_Games r inner join Digital_Video_Games u on u.product_id=r.product_id;

```
Time taken: 0.197 seconds
hive>   select r.product_id,Average_rating_by_customer_for_Digital_Video_Games,Average_rating_by_customer_for_Video_Games from Video_Games r inner join Digita
l_Video_Games u on u.product_id=r.product_id;
Query ID = hadoop_20200406022255_143d4705-5b0e-4e92-afee-f81f3f889ddd
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1586115072805_0022)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     17         17        0        0       0       0
Map 4 .......... container     SUCCEEDED     17         17        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     10         10        0        0       0       0
Reducer 3 ...... container     SUCCEEDED     10         10        0        0       0       0
Reducer 5 ...... container     SUCCEEDED     10         10        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 294.15 s
--------------------------------------------------------------------------------
OK
B0047T7MEW      3.69    4.5
B00NBBME0Y      5.0     3.47
B004YNII9Y      2.82    4.0
B00B4WVTUS      4.64    5.0
Time taken: 302.521 seconds, Fetched: 4 row(s)
hive>
```

ort MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

We can see that these products are common in both categories and they do not get same average ratings.

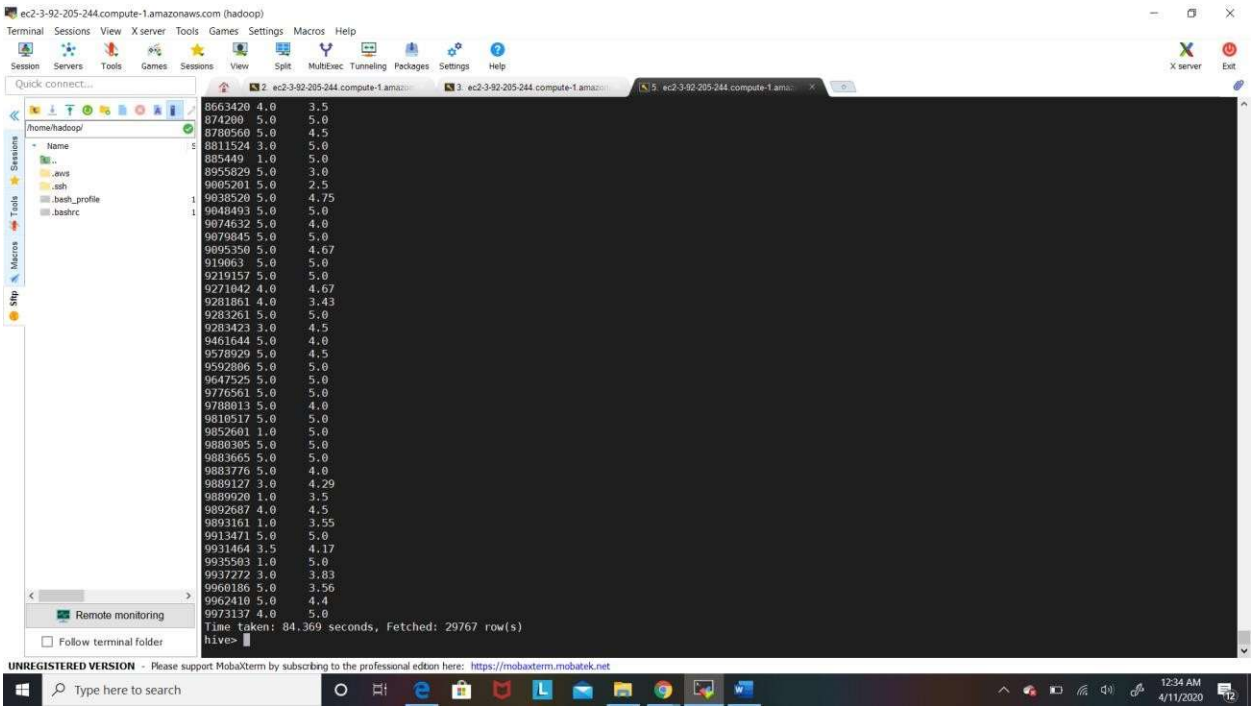
iv.You should cover additional questions and not limit yourself to the above questions

List of customers who have given reviews for products in both Digital_Video_Games and Video_Games category and their ratings in both categories.

Query-

```
create view video_games as
 select customer_id,product_category,round(avg(star_rating),2) as
Average_rating_by_customer_for_video_games_products from
amazon_review.filtered_reviews where product_category='Video_Games' and year>=2005
group by customer_id,product_category;

 create view Digital_Video_Games as
 select customer_id,product_category,round(avg(star_rating),2) as
Average_rating_by_customer_for_Digital_video_games_products from
amazon_review.filtered_reviews where product_category='Digital_Video_Games' and
year>=2005 group by customer_id,product_category;

select
r.customer_id,Average_rating_by_customer_for_Digital_video_games_products,Average_ratin
g_by_customer_for_video_games_products from Digital_Video_Games r inner join
video_games u on u.customer_id=r.customer_id;
```

Output-



3. You should demonstrate your ability to use Hive advanced functions:

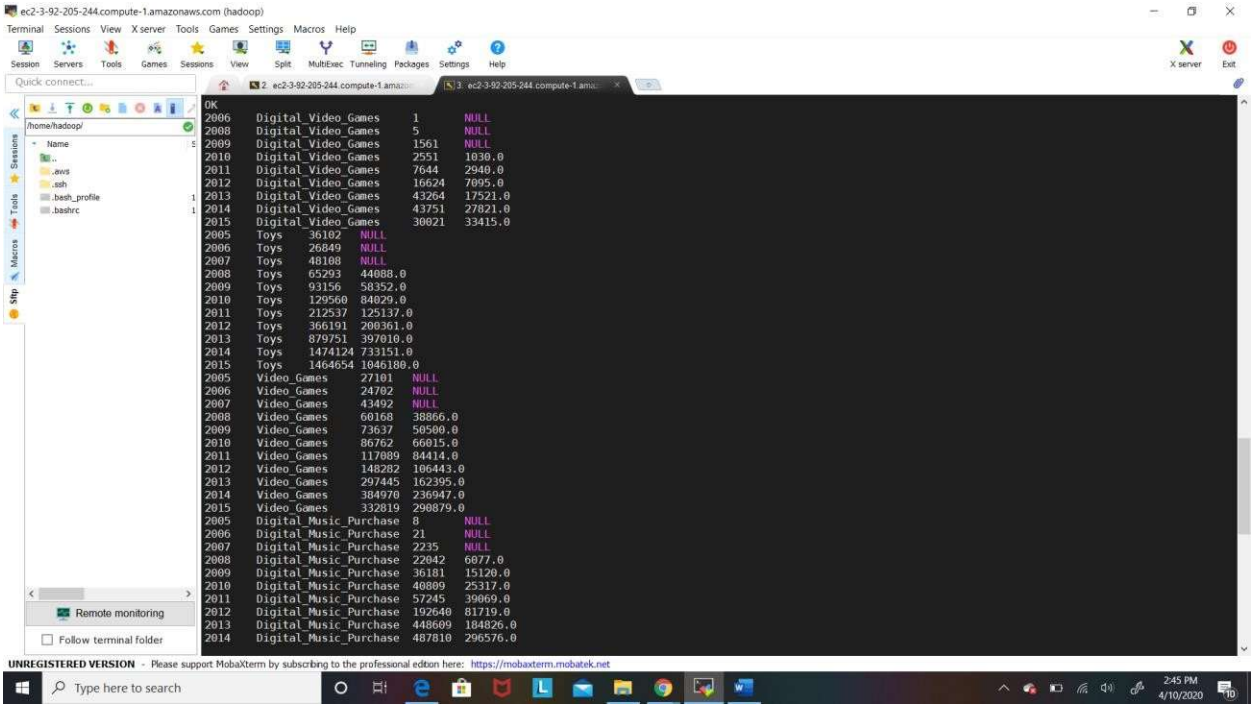
i. Window functions: moving average, rank, aggregation functions using relevant ordering and partitioning

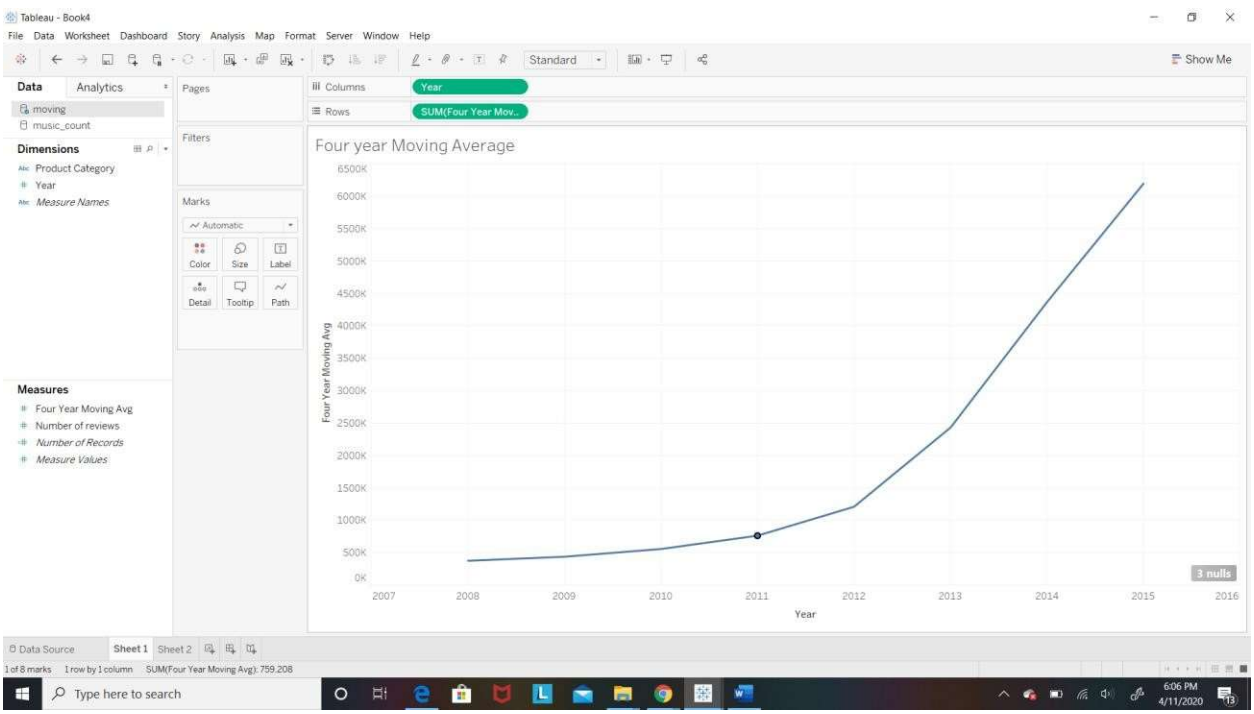
Calculating four year Moving average(current year and previous three years) based on number of reviews per product category over the years.

Query-

```
select year,product_category,Number_of_reviews,(case when row_number() over (Partition by
product_category order by year) > 3
then round(AVG(Number_of_reviews) OVER (PARTITION BY product_category order by year ROWS 3
PRECEDING))
end) as four_year_moving_avg from
(Select year,product_category,count(review_id) as Number_of_reviews,count(Distinct(customer_id)) as
Number_of_users,avg(star_rating) as average_review_stars,avg(length(review_body)) as
average_length_of_review
from amazon_review.filtered_reviews group by year,product_category order by product_category,year)
as x where year>=2005;
```

Output-





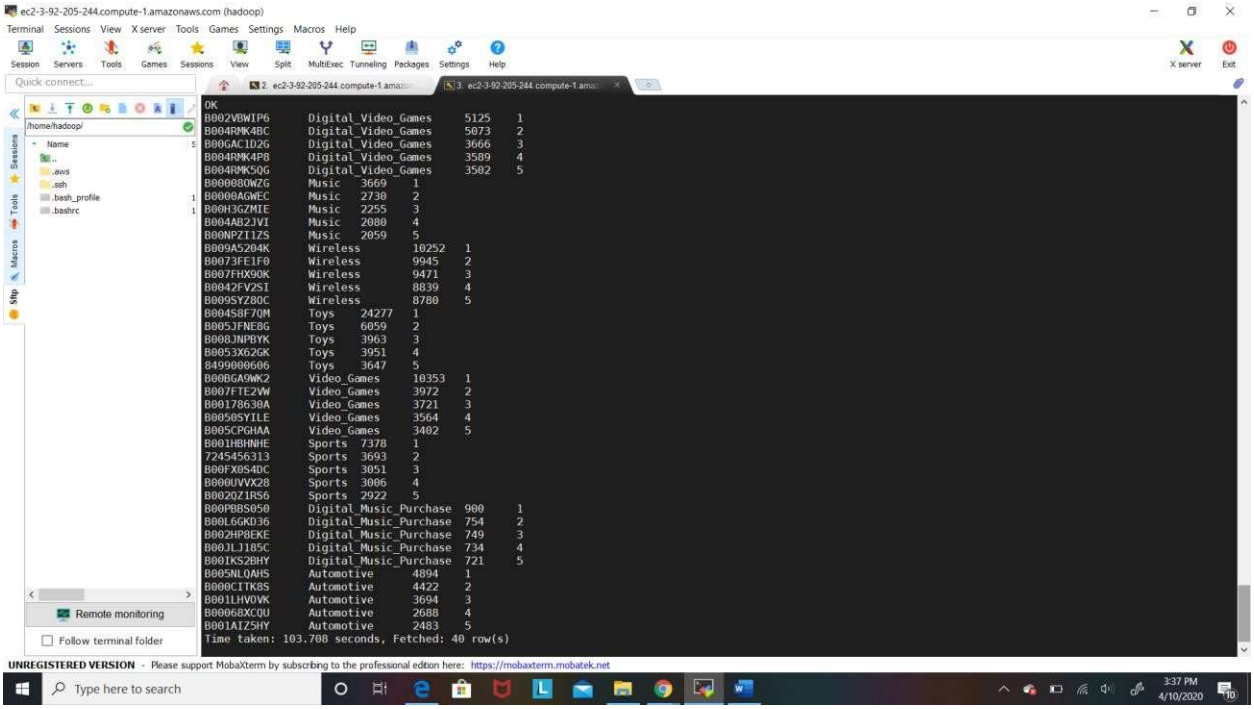
We can infer that Number of reviews are increasing over the years.

Ranking Top five products in each product categories based on number of reviews per product.

Query-

```
select product_id,product_category,Number_of_reviews,rank from(
select product_id,product_category,Number_of_reviews,rank() over (Partition by product_category
order by Number_of_reviews desc) as rank from
(Select product_id,product_category,count(review_id) as
Number_of_reviews,count(Distinct(customer_id)) as Number_of_users,avg(star_rating) as
average_review_stars,avg(length(review_body)) as average_length_of_review
from amazon_review.filtered_reviews group by product_category,product_id)as x)as z where rank<=5;
```

Output-



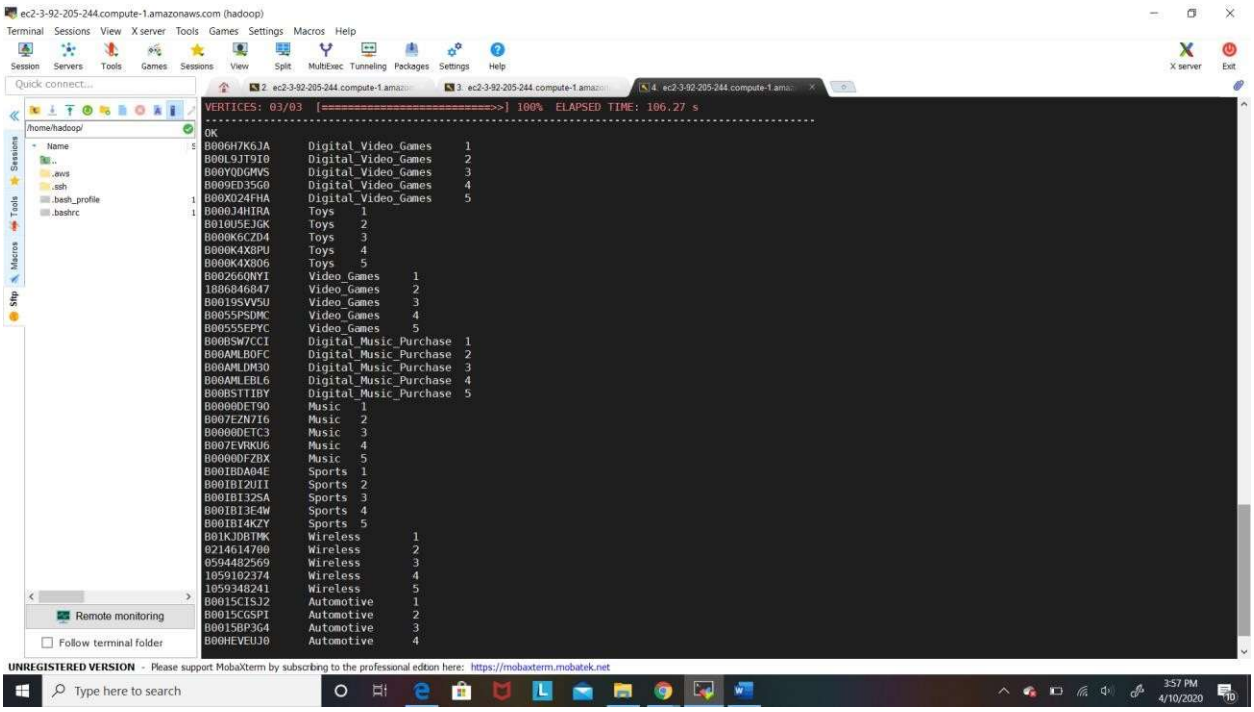
Ranking top five products in each category based on Average star rating for each product.

Query-

```
SELECT v.product_id,
      v.product_category,
      v.star_rank
FROM
   (SELECT z.product_id,
      z.product_category,z.avg_rating,
```

```
    Row_number()
    OVER (partition by z.product_category
    ORDER BY  z.avg_rating desc) AS star_rank
  FROM
    (SELECT product_id,
    product_category,
    avg(star_rating) AS avg_rating
    FROM amazon_review.filtered_reviews
    WHERE year>= 2005
    GROUP BY product_id,product_category)as z)as v
    WHERE v.star_rank<=5;
```

Output-



ii. Analytical Aggregate functions: percentile, min, max, average, standard deviation, correlation

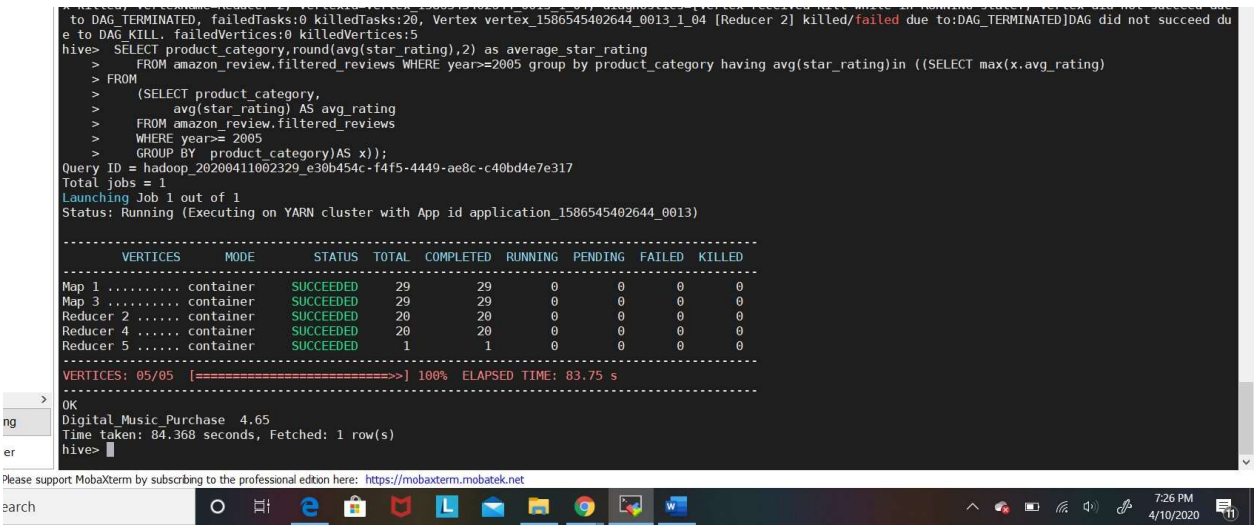
Using Max function to find out category which has got maximum average star rating.

Query-

```
SELECT product_category,round(avg(star_rating),2) as average_star_rating
```

FROM amazon_review.filtered_reviews WHERE year>=2005 group by product_category having avg(star_rating)in ((SELECT max(x.avg_rating)
FROM
(SELECT product_category,
avg(star_rating) AS avg_rating
FROM amazon_review.filtered_reviews
WHERE year>= 2005
GROUP BY product_category)AS x));

Output-



From this we can infer that **Digital music purchase category** has highest average star rating of **4.65.**

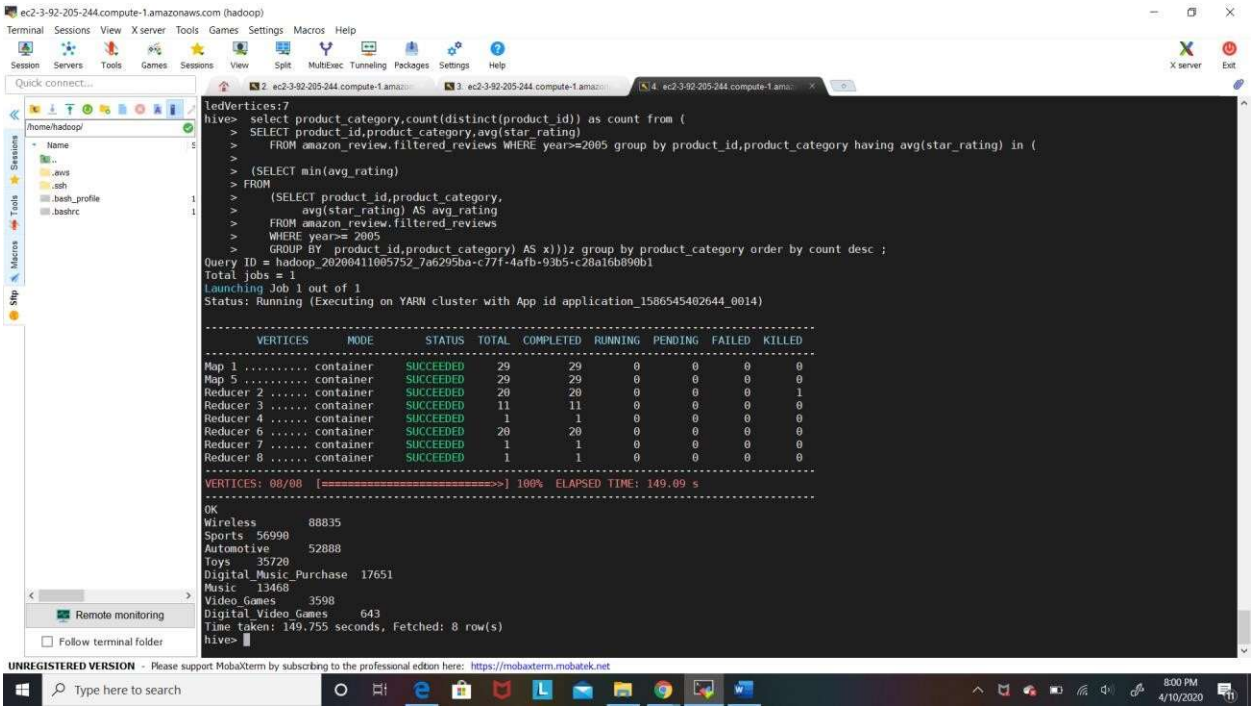
Using Minimum function to find out count of distinct products in each category that have got minimum average star rating.

Query-

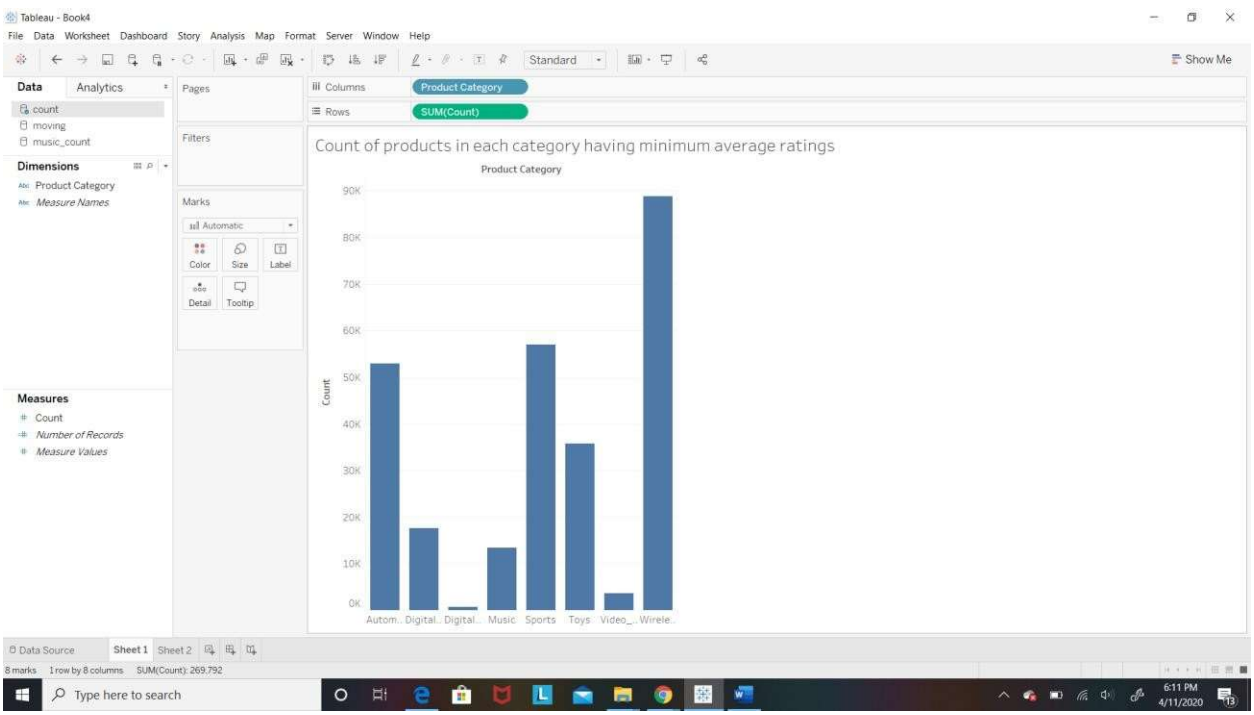
select product_category,count(distinct(product_id)) from (
SELECT product_id,product_category,avg(star_rating)
FROM amazon_review.filtered_reviews WHERE year>=2005 group by product_id,product_category having avg(star_rating) in (
(SELECT min(avg_rating)
FROM
(SELECT product_id,product_category,
avg(star_rating) AS avg_rating
FROM amazon_review.filtered_reviews
WHERE year>= 2005

GROUP BY product_id,product_category) AS x)))z group by product_category;

Output-



Visualization



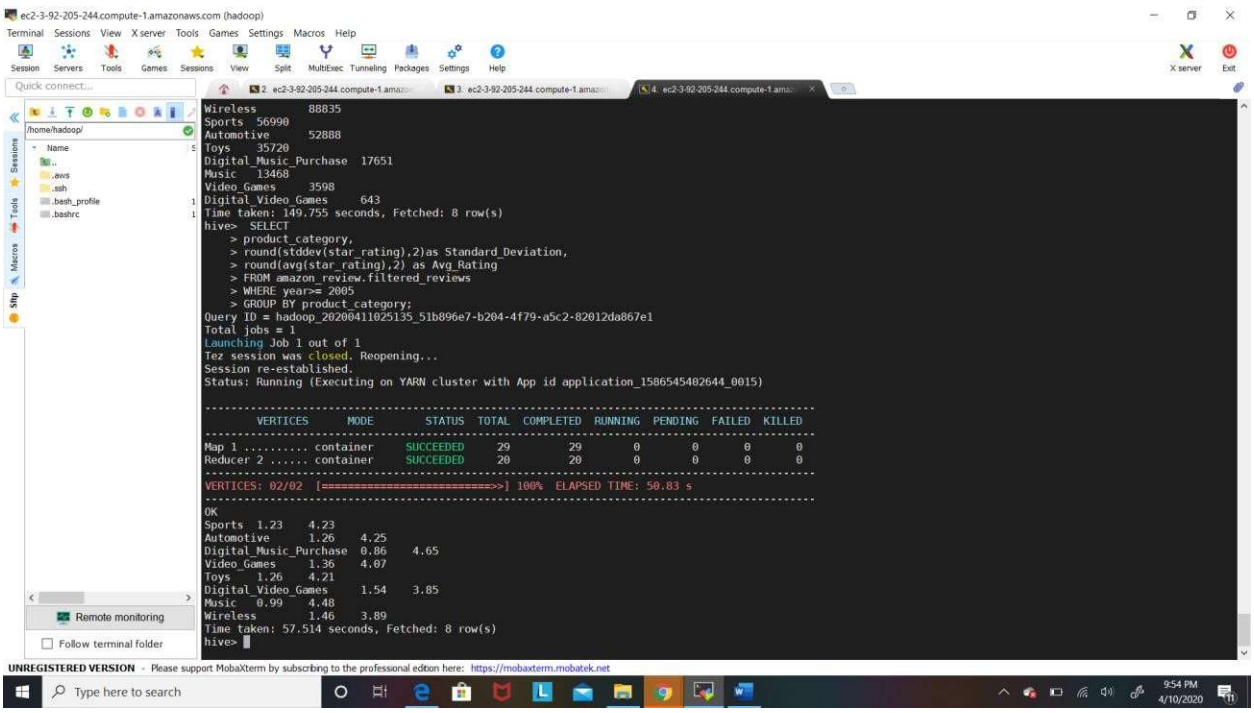
Interpretation- we can see that **Wireless category** has highest number of products that have received minimum average ratings.

Calculating Standard Deviation to analyze normal distribution of star rating of product categories.

Query-

```
SELECT
product_category,
round(stddev(star_rating),2)as Standard_Deviation,
round(avg(star_rating),2) as Avg_Rating
FROM amazon_review.filtered_reviews
WHERE year>= 2005
GROUP BY product_category;
```

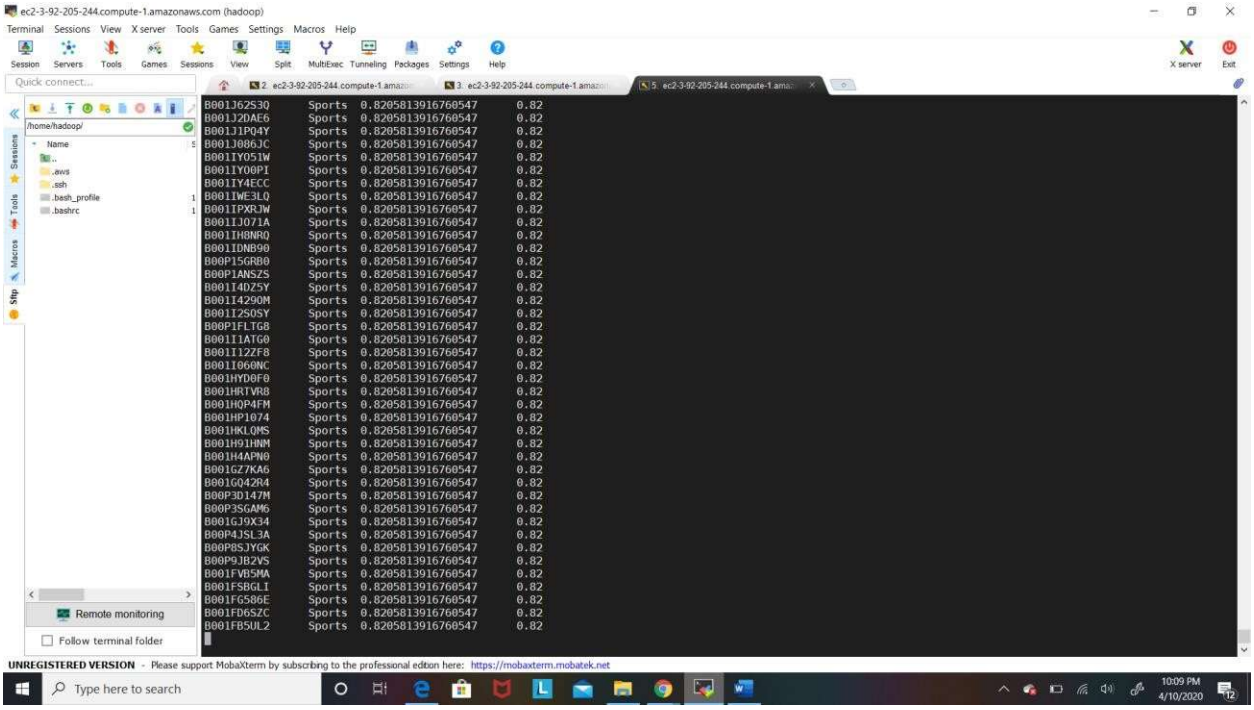
Output



We can see that Digital Music purchase category has least standard deviation which means it is least spread out in terms of average ratings from the mean.

Products having highest Percentile of star ratings given by customers:
Query-

```
SELECT y.product_id,
y.product_category,y.star_rank,
round(y.star_rank,
2) AS Rank_Percentile from
(SELECT x.product_id,
x.product_category,
PERCENT_RANK()
OVER (partition by x.product_category
ORDER BY x.avg_rating desc) AS star_rank
FROM
(SELECT product_id,
product_category,
avg(star_rating) AS avg_rating
FROM amazon_review.filtered_reviews
WHERE year>= 2005
GROUP BY product_id,product_category)as x)as y order by y.star_rank
desc;
```



References- https://towardsdatascience.com/converting-thumbs-up-thumbs-down-to-percentiles-with-skewness-intact-5ee70574a694

https://dzone.com/articles/100-shades-of-grey

Database management lectures and Advanced sql slides and code for Moving average.