



Amazon Electronics Products Analysis

By,

Sushmanth Sagala

Hadoop Project – UpX Academy



Project Information



- **Domain:** Retail
- **Technology use:** Hadoop, Hive
- **Dataset:** <http://jmcauley.ucsd.edu/data/amazon/links.html>
- The Amazon Product data set to be considered for this project is from Amazon, a well-known e-commerce giant. This is a very interesting project to get insights based analysis of product sales based on reviews and ratings.
- This dataset contains product reviews including 142.8 million reviews spanning May 1996 - July 2014. This data set includes reviews (ratings, text, helpfulness votes). From the various categories the Electronics category is chosen for the analysis. The size of reviews data set is 1.5 GB and ratings data set is 320 MB.



Data Description



➤ Review:

- reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- asin - ID of the product, e.g. 0000013714
- reviewerName - name of the reviewer
- helpful - helpfulness rating of the review, e.g. 2/3
- reviewText - text of the review
- overall - rating of the product
- summary - summary of the review
- unixReviewTime - time of the review (unix time)
- reviewTime - time of the review (raw)

➤ Rating:

- user - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- Item - ID of the product, e.g. 0000013714
- Rating - rating of the product
- Timestamp - time of the review (unix time)



Business Questions

► Hadoop / Hive

- Understand the Dataset
- Top electronic item that was rated above 3.0 over the period of 8 years [2006-2014] (EG. Year 2006 - 100 items bought received all the rating above 3.0 - show top 1 item out of 100; sample data 2006 ,“ASIN”, ‘Review Count’)
- Worst review a product received in a given year between 2006 to 2014 (Eg. 2006, ASIN, Review Count (Least number of reviews and below 2.0 rating))
- Maximum number of reviews given by same user in a year
- Least Helpful reviews per year per products (helpful percentage between 1% and 30 %)
- Most Helpful reviews Per year Per Products (Helpful percentage >75%)
- Growth of review comments on products/year
- Visualization of results observe

Q1. Understand the Dataset

Reviews dataset:

- Reviews dataset brought to HDFS as a Hive table using org.apache.hive.hcatalog.data.JsonSerDe library to load Json data.
 - CREATE TABLE amazon_reviews (reviewerID string, asin string, reviewerName string, helpful array<int>, reviewText string, overall float, summary string, unixReviewTime int, reviewTime string) ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe' STORED AS TEXTFILE;
 - LOAD DATA LOCAL INPATH 'data/Reviews/Electronics_5.json' OVERWRITE INTO TABLE amazon_reviews;
- Partitioned reviews data based on review year and clustered the data based on review rating.
 - CREATE TABLE reviews (reviewerID string, asin string, reviewerName string, helpful array<int>, reviewText string, overall float, summary string, unixReviewTime int, reviewTime string) PARTITIONED BY (reviewYear int) CLUSTERED BY (overall) INTO 5 BUCKETS ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
 - INSERT INTO TABLE reviews PARTITION(reviewYear) SELECT reviewerID,asin,reviewerName,helpful,reviewText,overall,summary,unixReviewTime,reviewTime,year(from_unixtime(unix_timestamp(reviewTime, 'MM dd, yyyy')))) FROM amazon_reviews;


```
cloudera@quickstart:~/Hadoop/Project/AmazonElectronics
Time taken for load dynamic partitions : 45331
Loading partition {reviewyear=2003}
Loading partition {reviewyear=2004}
Loading partition {reviewyear=2014}
Loading partition {reviewyear=2002}
Loading partition {reviewyear=2001}
Loading partition {reviewyear=2005}
Loading partition {reviewyear=2006}
Loading partition {reviewyear=2007}
Loading partition {reviewyear=2009}
Loading partition {reviewyear=1999}
Loading partition {reviewyear=2010}
Loading partition {reviewyear=2008}
Loading partition {reviewyear=2000}
Loading partition {reviewyear=2012}
Loading partition {reviewyear=2013}
Loading partition {reviewyear=2011}
Time taken for adding to write entity : 44
Partition electronics.reviews{reviewyear=1999} stats: [numFiles=5, numRows=72, totalSize=77350, rawDataSize=77278]
Partition electronics.reviews{reviewyear=2000} stats: [numFiles=5, numRows=817, totalSize=971355, rawDataSize=970538]
Partition electronics.reviews{reviewyear=2001} stats: [numFiles=5, numRows=1609, totalSize=1819209, rawDataSize=1817600]
Partition electronics.reviews{reviewyear=2002} stats: [numFiles=5, numRows=2315, totalSize=2787153, rawDataSize=2784838]
Partition electronics.reviews{reviewyear=2003} stats: [numFiles=5, numRows=3547, totalSize=4199064, rawDataSize=4195517]
Partition electronics.reviews{reviewyear=2004} stats: [numFiles=5, numRows=5159, totalSize=6495688, rawDataSize=6490529]
Partition electronics.reviews{reviewyear=2005} stats: [numFiles=5, numRows=9638, totalSize=11830267, rawDataSize=11820629]
Partition electronics.reviews{reviewyear=2006} stats: [numFiles=5, numRows=15447, totalSize=17890345, rawDataSize=17874898]
Partition electronics.reviews{reviewyear=2007} stats: [numFiles=5, numRows=35976, totalSize=32913238, rawDataSize=32877262]
Partition electronics.reviews{reviewyear=2008} stats: [numFiles=5, numRows=49872, totalSize=48537103, rawDataSize=48487231]
Partition electronics.reviews{reviewyear=2009} stats: [numFiles=5, numRows=70666, totalSize=72259502, rawDataSize=72188836]
Partition electronics.reviews{reviewyear=2010} stats: [numFiles=5, numRows=103797, totalSize=102676321, rawDataSize=102572524]
Partition electronics.reviews{reviewyear=2011} stats: [numFiles=5, numRows=173395, totalSize=158977173, rawDataSize=158803778]
Partition electronics.reviews{reviewyear=2012} stats: [numFiles=5, numRows=282942, totalSize=224637670, rawDataSize=224354728]
Partition electronics.reviews{reviewyear=2013} stats: [numFiles=5, numRows=592748, totalSize=357932828, rawDataSize=357340080]
Partition electronics.reviews{reviewyear=2014} stats: [numFiles=5, numRows=341188, totalSize=194132185, rawDataSize=193790997]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 6 Reduce: 5 Cumulative CPU: 149.94 sec HDFS Read: 1479063254 HDFS Write: 1238140504 SUCCESS
Total MapReduce CPU Time Spent: 2 minutes 29 seconds 940 msec
OK
Time taken: 806.684 seconds
hive>
```

Sample output screenshot

Q1. Understand the Dataset

Ratings dataset:

- ▶ Ratings dataset brought to HDFS as a Hive table.
 - ▶ CREATE TABLE amazon_ratings (user string, item string, rating float, timestamp int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
 - ▶ LOAD DATA LOCAL INPATH 'data/Ratings/ratings_Electronics.csv' OVERWRITE INTO TABLE amazon_ratings;
- ▶ Partitioned rating data based on rating year and clustered the data based on rating.
 - ▶ CREATE TABLE ratings (user string, item string, rating float, timestamp int) PARTITIONED BY (ratingYear int) CLUSTERED BY (rating) INTO 5 BUCKETS ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
 - ▶ INSERT INTO TABLE ratings PARTITION(ratingYear) SELECT user,item,rating,timestamp,year(date_add(from_unixtime(timestamp),1)) FROM amazon_ratings;

```
cloudera@quickstart:~/Hadoop/Project/AmazonElectronics
Loading partition {ratingyear=2014}
Loading partition {ratingyear=2000}
Loading partition {ratingyear=2002}
Loading partition {ratingyear=2003}
Loading partition {ratingyear=2010}
Loading partition {ratingyear=2013}
Loading partition {ratingyear=2011}
Loading partition {ratingyear=2012}
Loading partition {ratingyear=2006}
Loading partition {ratingyear=2007}
Loading partition {ratingyear=2005}
Loading partition {ratingyear=2008}
Loading partition {ratingyear=2004}
Loading partition {ratingyear=1998}
Loading partition {ratingyear=2009}
Loading partition {ratingyear=1999}
Time taken for adding to write entity : 6
Partition electronics.ratings{ratingyear=1998} stats: [numFiles=5, numRows=4, totalSize=158, rawDataSize=154]
Partition electronics.ratings{ratingyear=1999} stats: [numFiles=5, numRows=1212, totalSize=48148, rawDataSize=46936]
Partition electronics.ratings{ratingyear=2000} stats: [numFiles=5, numRows=9214, totalSize=366009, rawDataSize=356795]
Partition electronics.ratings{ratingyear=2001} stats: [numFiles=5, numRows=14753, totalSize=591914, rawDataSize=577161]
Partition electronics.ratings{ratingyear=2002} stats: [numFiles=5, numRows=18648, totalSize=759688, rawDataSize=741040]
Partition electronics.ratings{ratingyear=2003} stats: [numFiles=5, numRows=23117, totalSize=941576, rawDataSize=918459]
Partition electronics.ratings{ratingyear=2004} stats: [numFiles=5, numRows=31245, totalSize=1272864, rawDataSize=1241619]
Partition electronics.ratings{ratingyear=2005} stats: [numFiles=5, numRows=56311, totalSize=2293865, rawDataSize=2237554]
Partition electronics.ratings{ratingyear=2006} stats: [numFiles=5, numRows=86659, totalSize=3530215, rawDataSize=3443556]
Partition electronics.ratings{ratingyear=2007} stats: [numFiles=5, numRows=193430, totalSize=7879722, rawDataSize=7686292]
Partition electronics.ratings{ratingyear=2008} stats: [numFiles=5, numRows=238521, totalSize=9716299, rawDataSize=9477778]
Partition electronics.ratings{ratingyear=2009} stats: [numFiles=5, numRows=325380, totalSize=13254128, rawDataSize=12928748]
Partition electronics.ratings{ratingyear=2010} stats: [numFiles=5, numRows=475626, totalSize=19374438, rawDataSize=18898812]
Partition electronics.ratings{ratingyear=2011} stats: [numFiles=5, numRows=783503, totalSize=31916580, rawDataSize=31133077]
Partition electronics.ratings{ratingyear=2012} stats: [numFiles=5, numRows=1231673, totalSize=50183198, rawDataSize=48951525]
Partition electronics.ratings{ratingyear=2013} stats: [numFiles=5, numRows=2626582, totalSize=107021027, rawDataSize=104394445]
Partition electronics.ratings{ratingyear=2014} stats: [numFiles=5, numRows=1708604, totalSize=69616668, rawDataSize=67908064]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2 Reduce: 5 Cumulative CPU: 238.49 sec HDFS Read: 318802807 HDFS Write: 318770675 SUCCESS
Total MapReduce CPU Time Spent: 3 minutes 58 seconds 490 msec
OK
Time taken: 322.419 seconds
hive>
```

Sample output screenshot

Business Questions Answered

- Q2. Top electronic item that was rated above 3.0 over the period of 8 years [2006-2014] (EG. Year 2006 - 100 items bought received all the rating above 3.0 - show top 1 item out of 100; sample data 2006 , "ASIN", 'Review Count')
 - To determine the Top electronics item rated above 3.0 against each year [2006-2014].
 - Used outer join as all reviewers will be present in ratings dataset. Right outer join is used as ratings is bigger table.
 - Grouped all the Items against each year with avg rating greater than 3.0 and choosing the item with highest review count against each year [2006-2014].

```
SELECT r.ratingYear,r.item,r.review_count
```

```
FROM (SELECT rt.ratingYear,rt.item,rt.review_count,ROW_NUMBER() OVER(PARTITION BY rt.ratingYear ORDER BY rt.review_count DESC) AS item_rank
```

```
FROM (SELECT rat.ratingYear,rat.item,AVG(rat.rating) AS avg_rating,COUNT(rev.reviewerID) AS review_count
```

```
FROM (SELECT reviewYear,asin,reviewerID FROM reviews WHERE reviewYear >= 2006 AND reviewYear <= 2014) rev
```

```
LEFT OUTER JOIN (SELECT ratingYear,item,user,rating FROM ratings WHERE ratingYear >= 2006 AND ratingYear <= 2014) rat
```

```
ON rev.asin = rat.item AND rev.reviewerID = rat.user
```

```
GROUP BY rat.ratingYear,rat.item ) rt
```

```
WHERE avg_rating > CAST(3.0 AS FLOAT) ) r
```

```
WHERE r.item_rank = 1;
```

Business Questions Answered

- Q3. Worst review a product received in a given year between 2006 to 2014 (Eg. 2006, ASIN, Review Count (Least number of reviews and below 2.0 rating)
 - To determine the worst electronics item rated below 2.0 against each year [2006-2014].
 - Used outer join as all reviewers will be present in ratings dataset. Right outer join is used as ratings is bigger table.
 - Grouped all the Items against each year with avg rating lesser than 2.0 and choosing the item with least review count against each year [2006-2014].

```
SELECT r.ratingYear,r.item,r.review_count
```

```
FROM (SELECT rt.ratingYear,rt.item,rt.review_count,ROW_NUMBER() OVER(PARTITION BY rt.ratingYear  
ORDER BY rt.review_count ASC) AS item_rank
```

```
FROM (SELECT rat.ratingYear,rat.item,AVG(rat.rating) AS avg_rating,COUNT(rev.reviewerID) AS  
review_count
```

```
FROM (SELECT reviewYear,asin,reviewerID FROM reviews WHERE reviewYear >= 2006 AND reviewYear <=  
2014) rev
```

```
RIGHT OUTER JOIN (SELECT ratingYear,item,user,rating FROM ratings WHERE ratingYear >= 2006 AND  
ratingYear <= 2014) rat
```

```
ON rev.asin = rat.item AND rev.reviewerID = rat.user
```

```
GROUP BY rat.ratingYear,rat.item ) rt
```

```
WHERE avg_rating < CAST(2.0 AS FLOAT) ) r
```

```
WHERE r.item_rank = 1;
```

```
cloudera@quickstart:~/Hadoop/Project/AmazonElectronics
2018-06-10 08:56:07,610 Stage-2 map = 100%, reduce = 83%, Cumulative CPU 52.51 sec
2018-06-10 08:56:10,755 Stage-2 map = 100%, reduce = 95%, Cumulative CPU 55.59 sec
2018-06-10 08:56:12,915 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 57.07 sec
MapReduce Total cumulative CPU time: 57 seconds 70 msec
Ended Job = job_1528528998925_0035
Launching Job 3 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1528528998925_0036, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1528528998925_0036/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1528528998925_0036
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2018-06-10 08:56:34,369 Stage-3 map = 0%, reduce = 0%
2018-06-10 08:56:51,369 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 11.04 sec
2018-06-10 08:57:06,408 Stage-3 map = 100%, reduce = 70%, Cumulative CPU 18.39 sec
2018-06-10 08:57:09,565 Stage-3 map = 100%, reduce = 93%, Cumulative CPU 22.12 sec
2018-06-10 08:57:11,697 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 24.3 sec
MapReduce Total cumulative CPU time: 24 seconds 300 msec
Ended Job = job_1528528998925_0036
MapReduce Jobs Launched:
Stage-Stage-1: Map: 7 Reduce: 6 Cumulative CPU: 357.09 sec HDFS Read: 1522569315 HDFS Write: 101301379 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 57.07 sec HDFS Read: 101309202 HDFS Write: 23098966 SUCCESS
Stage-Stage-3: Map: 1 Reduce: 1 Cumulative CPU: 24.3 sec HDFS Read: 23106125 HDFS Write: 182 SUCCESS
Total MapReduce CPU Time Spent: 7 minutes 18 seconds 460 msec
OK
2006 B000CS7U1C 89
2007 B000BKJZ9Q 192
2008 B000LRMS66 270
2009 B0002L5R78 272
2010 B001T9NUJE 260
2011 B0019EHU8G 527
2012 B0019EHU8G 1010
2013 B007WTAJTO 2678
2014 B00DR0PDNE 2039
Time taken: 750.15 seconds, Fetched: 9 row(s)
hive>
```

Sample output screenshot

Business Questions Answered

- Q4. Maximum number of reviews given by same user in a year
 - To determine the reviewer by whom maximum number of reviews given in a year.
 - Used outer join as all reviewers will be present in ratings dataset. Right outer join is used as ratings is bigger table.
 - Grouped all the reviewers against each year to find the highest reviews given.

```
SELECT rev.reviewYear,rev.reviewerID,COUNT(rev.reviewerID) AS review_count
FROM reviews rev
RIGHT OUTER JOIN ratings rat
ON rev.asin = rat.item AND rev.reviewerID = rat.user
GROUP BY rev.reviewYear,rev.reviewerID
ORDER BY review_count DESC LIMIT 1;
```



```
cloudera@quickstart:~/Hadoop/Project/AmazonElectronics
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1528528998925_0038, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1528528998925_0038/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1528528998925_0038
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-06-10 09:07:42,857 Stage-2 map = 0%, reduce = 0%
2018-06-10 09:07:56,594 Stage-2 map = 10%, reduce = 0%, Cumulative CPU 7.87 sec
2018-06-10 09:07:59,792 Stage-2 map = 38%, reduce = 0%, Cumulative CPU 10.95 sec
2018-06-10 09:08:02,960 Stage-2 map = 57%, reduce = 0%, Cumulative CPU 13.83 sec
2018-06-10 09:08:06,077 Stage-2 map = 67%, reduce = 0%, Cumulative CPU 17.05 sec
2018-06-10 09:08:09,245 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 20.57 sec
2018-06-10 09:08:23,321 Stage-2 map = 100%, reduce = 70%, Cumulative CPU 28.62 sec
2018-06-10 09:08:26,453 Stage-2 map = 100%, reduce = 95%, Cumulative CPU 31.79 sec
2018-06-10 09:08:27,534 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 32.7 sec
MapReduce Total cumulative CPU time: 32 seconds 700 msec
Ended Job = job_1528528998925_0038
Launching Job 3 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1528528998925_0039, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1528528998925_0039/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1528528998925_0039
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2018-06-10 09:08:50,954 Stage-3 map = 0%, reduce = 0%
2018-06-10 09:09:06,081 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 9.7 sec
2018-06-10 09:09:19,959 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 14.82 sec
MapReduce Total cumulative CPU time: 14 seconds 820 msec
Ended Job = job_1528528998925_0039
MapReduce Jobs Launched:
Stage-Stage-1: Map: 7 Reduce: 7 Cumulative CPU: 277.69 sec HDFS Read: 1557064802 HDFS Write: 45917504 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 32.7 sec HDFS Read: 45923907 HDFS Write: 20365338 SUCCESS
Stage-Stage-3: Map: 1 Reduce: 1 Cumulative CPU: 14.82 sec HDFS Read: 20370481 HDFS Write: 24 SUCCESS
Total MapReduce CPU Time Spent: 5 minutes 25 seconds 210 msec
OK
2012 A2AY4YUOX2N1BQ 126
Time taken: 563.439 seconds, Fetched: 1 row(s)
```

Sample output screenshot

Business Questions Answered

- Q5. Least Helpful reviews per year per products (helpful percentage between 1% and 30 %)
 - To determine the Item per year with least helpful reviews within 1% and 30%.
 - Used outer join as all reviewers will be present in ratings dataset. Right outer join is used as ratings is bigger table.
 - Grouped all the Items against each year with avg helpful percentage between 1% and 30%, choosing the item with least helpful reviews against each year.

```
SELECT r.ratingYear,r.item,r.avg_helpful_perc
FROM (SELECT rt.ratingYear,rt.item,rt.avg_helpful_perc,ROW_NUMBER() OVER(PARTITION BY
rt.ratingYear ORDER BY rt.avg_helpful_perc ASC) AS helpful_perc_rank
FROM (SELECT rat.ratingYear,rat.item,AVG((helpful[0] / helpful[1]) * 100) AS avg_helpful_perc
FROM reviews rev
RIGHT OUTER JOIN ratings rat
ON rev.asin = rat.item AND rev.reviewerID = rat.user
GROUP BY rat.ratingYear,rat.item ) rt
WHERE rt.avg_helpful_perc >= CAST(1.0 AS FLOAT) AND avg_helpful_perc <= CAST(30.0 AS FLOAT)
) r
WHERE r.helpful_perc_rank = 1;
```

Business Questions Answered

- Q6. Most Helpful reviews Per year Per Products (Helpful percentage >75%)
 - To determine the Item per year with most helpful reviews above 75%.
 - Used outer join as all reviewers will be present in ratings dataset. Right outer join is used as ratings is bigger table.
 - Grouped all the Items against each year with avg helpful percentage above 75%, choosing the item with most helpful reviews against each year.

```
SELECT r.ratingYear,r.item,r.avg_helpful_perc  
  
FROM (SELECT rt.ratingYear,rt.item,rt.avg_helpful_perc,ROW_NUMBER() OVER(PARTITION BY  
rt.ratingYear ORDER BY rt.avg_helpful_perc DESC) AS helpful_perc_rank  
  
FROM (SELECT rat.ratingYear,rat.item,AVG((helpful[0] / helpful[1]) * 100) AS avg_helpful_perc  
FROM reviews rev  
RIGHT OUTER JOIN ratings rat  
ON rev.asin = rat.item AND rev.reviewerID = rat.user  
GROUP BY rat.ratingYear,rat.item ) rt  
WHERE rt.avg_helpful_perc > CAST(75.0 AS FLOAT) ) r  
WHERE r.helpful_perc_rank = 1;
```

```

cloudera@quickstart:~
2018-06-10 09:45:25,067 Stage-2 map = 48%, reduce = 0%, Cumulative CPU 25.61 sec
2018-06-10 09:45:31,321 Stage-2 map = 57%, reduce = 0%, Cumulative CPU 31.61 sec
2018-06-10 09:45:34,443 Stage-2 map = 67%, reduce = 0%, Cumulative CPU 34.62 sec
2018-06-10 09:45:37,597 Stage-2 map = 75%, reduce = 0%, Cumulative CPU 37.72 sec
2018-06-10 09:45:40,744 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 40.89 sec
2018-06-10 09:45:55,731 Stage-2 map = 100%, reduce = 67%, Cumulative CPU 47.22 sec
2018-06-10 09:45:58,867 Stage-2 map = 100%, reduce = 76%, Cumulative CPU 51.14 sec
2018-06-10 09:46:02,040 Stage-2 map = 100%, reduce = 92%, Cumulative CPU 54.97 sec
2018-06-10 09:46:03,133 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 56.56 sec
MapReduce Total cumulative CPU time: 56 seconds 560 msec
Ended Job = job_1528528998925_0044
Launching Job 3 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1528528998925_0045, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1528528998925_0045/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1528528998925_0045
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2018-06-10 09:46:24,683 Stage-3 map = 0%, reduce = 0%
2018-06-10 09:46:35,380 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 4.28 sec
2018-06-10 09:46:48,286 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 9.08 sec
MapReduce Total cumulative CPU time: 9 seconds 80 msec
Ended Job = job_1528528998925_0045
MapReduce Jobs Launched:
Stage-Stage-1: Map: 7 Reduce: 7 Cumulative CPU: 380.73 sec HDFS Read: 1557073972 HDFS Write: 109180513 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 56.56 sec HDFS Read: 109188187 HDFS Write: 130446 SUCCESS
Stage-Stage-3: Map: 1 Reduce: 1 Cumulative CPU: 9.08 sec HDFS Read: 137918 HDFS Write: 515 SUCCESS
Total MapReduce CPU Time Spent: 7 minutes 26 seconds 370 msec
OK
2000 B00000JD4T 11.11111111111111
2001 B00004TL5W 6.666666666666667
2002 B00006B9HP 5.88235294117647
2003 B00006DY6J 3.8461538461538463
2004 B0002EXJ8Y 6.666666666666667
2005 B000067O7K 5.88235294117647
2006 B0000W4UIS 1.1494252873563218
2007 B000VE7GQQ 4.411764705882353
2008 B00009R9EQ 1.0869565217391304
2009 B0007ZA15M 1.9607843137254901
2010 B0036A4Y9O 1.7543859649122806
2011 B00415E7FE 2.0833333333333333
2012 B007D4ZDJO 2.631578947368421
2013 B00747WW9E 1.0204081632653061
2014 B00AqDG96K 3.3333333333333335
Time taken: 1399.296 seconds, Fetched: 15 row(s)
hive>

```

Sample output screenshot

Business Questions Answered

➤ Q7. Growth of review comments on products/year

- To determine the growth of the review comments against each product every year.
- Transposed review count of each year against every product using Map function and group by.
- Grouped each product with a map of review counts against each year.

```
SELECT r.asin,collect_list(r.year_count_map) AS Year_count
FROM (SELECT rev.asin,map(rev.reviewYear,COUNT(rev.reviewerID)) AS year_count_map
FROM reviews rev
GROUP BY rev.reviewYear,rev.asin) r
GROUP BY r.asin;
```



```
cloudera@quickstart:~$
B00003WGP5 [[{2002:3},{2007:6},{2012:2},{2003:1},{2008:2},{2013:2},{2001:6},{2006:2},{2004:1},{2014:1},{2000:5},{2005:4}]]
B000068P8R [[{2005:2},{2010:1},{2004:1},{2009:1},{2014:5},{2003:1},{2013:26},{2006:2},{2011:2},{2002:1},{2007:1},{2012:6}]]
B00001W0DD [[{2000:1},{2005:3},{2010:1},{2004:1},{2009:3},{2014:2},{2007:1},{2003:1},{2013:1},{2001:1},{2006:2},{2011:1}]]
B00006B9HJ [[{2006:2},{2011:4},{2005:1},{2010:3},{2003:5},{2008:5},{2013:1},{2004:4},{2009:6},{2014:1},{2002:4},{2007:4}]]
B00004SABJ [[{2008:3},{2013:17},{2000:1},{2010:5},{2001:1},{2006:1},{2011:6},{2004:1},{2009:6},{2014:6},{2007:2},{2012:9}]]
B000058EGT [[{2004:2},{2009:2},{2005:1},{2010:1},{2001:3},{2006:1},{2003:3},{2008:3},{2013:2},{2002:2},{2007:2},{2012:2}]]
B00006H0L1 [[{2006:1},{2011:2},{2007:1},{2012:2},{2004:1},{2009:2},{2014:5},{2003:1},{2008:1},{2013:6},{2005:2},{2010:1}]]
B00005AW1H [[{2005:10},{2010:2},{2003:18},{2008:1},{2013:3},{2004:14},{2009:1},{2014:1},{2001:5},{2006:3},{2002:13},{2007:8}]]
B00007KDIVI [[{2005:69},{2010:17},{2006:77},{2011:7},{2004:44},{2009:16},{2014:5},{2007:95},{2012:7},{2003:45},{2008:67},{2013:11}]]
B00003006K [[{2000:2},{2010:4},{2004:1},{2009:4},{2014:1},{2002:1},{2007:2},{2012:3},{2003:2},{2008:4},{2013:10},{2011:2}]]
B00005TQ08 [[{2004:4},{2009:3},{2005:5},{2010:12},{2006:8},{2011:6},{2003:12},{2008:11},{2013:3},{2002:5},{2007:12},{2012:4}]]
B00000J061 [[{2000:4},{2005:11},{2004:6},{2009:1},{2014:4},{2003:3},{2008:2},{2013:4},{2002:4},{2007:5},{2001:4},{2006:7},{2011:1}]]
B00006B9CR [[{2002:1},{2007:10},{2012:9},{2004:3},{2009:17},{2014:6},{2005:9},{2010:16},{2006:7},{2011:16},{2003:2},{2008:16},{2013:17}]]
B00005T3XH [[{2006:1},{2011:3},{2002:2},{2007:3},{2012:12},{2005:2},{2010:4},{2003:1},{2008:4},{2013:9},{2004:1},{2009:3},{2014:1}]]
B00004VX15 [[{2003:2},{2008:11},{2013:6},{2000:1},{2010:1},{2007:7},{2012:11},{2004:1},{2009:8},{2014:4},{2001:2},{2006:2},{2011:5}]]
B00005853Z [[{2005:7},{2002:2},{2007:3},{2012:5},{2004:4},{2009:1},{2014:4},{2001:6},{2006:2},{2011:5},{2003:8},{2008:2},{2013:4}]]
B000058BCQ [[{2006:7},{2011:1},{2005:7},{2010:2},{2002:6},{2007:10},{2012:1},{2003:8},{2008:7},{2013:1},{2004:6},{2009:5},{2014:1}]]
B00004VX39 [[{2002:1},{2007:7},{2012:7},{2009:6},{2014:9},{2003:1},{2008:8},{2013:14},{2005:1},{2010:4},{2001:1},{2006:5},{2011:6}]]
B000062VUO [[{2004:6},{2009:5},{2014:35},{2006:21},{2011:41},{2003:6},{2008:21},{2013:52},{2005:8},{2010:22},{2002:6},{2007:31},{2012:45}]]
B00004ZCKV [[{2001:1},{2006:10},{2011:11},{2008:15},{2013:27},{2004:2},{2009:5},{2014:6},{2002:1},{2007:16},{2012:11},{2005:2},{2010:7}]]
B00004THCZ [[{2002:1},{2007:16},{2012:27},{2004:2},{2009:13},{2014:29},{2006:11},{2011:24},{2000:3},{2005:6},{2010:16},{2008:14},{2013:71}]]
B000065BFB [[{2005:4},{2010:20},{2006:10},{2011:27},{2004:4},{2009:19},{2014:35},{2002:1},{2007:21},{2012:45},{2003:4},{2008:20},{2013:74}]]
B00004SYNX [[{2004:2},{2009:3},{2014:1},{2003:1},{2013:5},{2000:2},{2010:2},{2001:2},{2006:2},{2011:6},{2002:2},{2007:2},{2012:3}]]
B00001W0DH [[{2003:2},{2008:6},{2013:4},{2006:5},{2011:7},{2004:4},{2009:7},{2014:2},{2002:1},{2007:5},{2012:7},{2005:3},{2010:3}]]
B00004Y2Q8 [[{2002:2},{2007:12},{2012:4},{2009:5},{2014:1},{2005:1},{2010:2},{2003:1},{2008:9},{2013:4},{2001:2},{2006:5},{2011:1}]]
B0000642RX [[{2003:6},{2008:4},{2013:12},{2002:1},{2007:7},{2012:6},{2004:6},{2009:11},{2014:10},{2006:4},{2011:35},{2005:2},{2010:15}]]
B000065BP9 [[{2003:1},{2008:51},{2013:22},{2002:1},{2007:44},{2012:19},{2006:17},{2011:61},{2004:6},{2009:69},{2014:3},{2005:13},{2010:96}]]
B00000JDF6 [[{2004:1},{2009:2},{2014:6},{2002:3},{2012:8},{2005:3},{2010:1},{2001:1},{2006:1},{2011:2},{2003:2},{2008:3},{2013:17}]]
B00005ARK3 [[{2003:39},{2013:2},{2005:13},{2010:1},{2001:27},{2006:11},{2011:2},{2004:17},{2009:2},{2014:1},{2002:58},{2007:5},{2012:1}]]
B00006HYPV [[{2004:8},{2009:14},{2014:6},{2006:14},{2011:4},{2005:13},{2010:4},{2002:1},{2007:35},{2012:5},{2003:2},{2008:24},{2013:7}]]
B00006B7HB [[{2006:5},{2011:6},{2005:4},{2010:5},{2002:12},{2007:3},{2012:5},{2004:5},{2009:1},{2014:5},{2003:13},{2008:2},{2013:4}]]
B00007056H [[{2005:7},{2010:24},{2006:13},{2011:27},{2002:1},{2007:18},{2012:25},{2004:5},{2009:11},{2014:38},{2003:4},{2008:17},{2013:80}]]
B00005T39Y [[{2002:1},{2007:26},{2012:11},{2005:13},{2010:13},{2004:2},{2009:22},{2014:19},{2006:9},{2011:24},{2003:1},{2008:20},{2013:24}]]
B00004Z5M1 [[{2002:3},{2007:18},{2012:33},{2006:9},{2011:26},{2004:3},{2009:17},{2014:43},{2005:4},{2010:16},{2003:2},{2008:14},{2013:88}]]
B00005T6GZ [[{2001:3},{2006:1},{2011:3},{2005:3},{2010:2},{2003:6},{2008:4},{2013:2},{2002:11},{2007:16},{2012:2},{2004:6},{2009:2}]]
B00005ATMB [[{2004:2},{2009:11},{2014:46},{2006:3},{2011:22},{2005:3},{2010:13},{2003:3},{2008:20},{2013:84},{2002:1},{2007:22},{2012:32}]]
B00005N6KG [[{2003:2},{2008:7},{2013:83},{2002:1},{2007:3},{2012:76},{2005:1},{2010:17},{2004:1},{2009:20},{2014:16},{2006:2},{2011:49}]]
B00001ZWXA [[{2004:3},{2009:9},{2014:19},{2002:1},{2007:4},{2012:25},{2000:1},{2005:1},{2010:6},{2008:10},{2013:40},{2001:2},{2006:1},{2011:14}]]
B00000K4KH [[{2000:2},{2005:1},{2010:9},{2002:3},{2007:3},{2012:6},{2006:1},{2011:5},{2004:1},{2009:7},{2014:8},{2003:1},{2008:4},{2013:19}]]
B00003006E [[{2008:16},{2013:5},{2001:3},{2006:4},{2011:6},{2004:4},{2009:15},{2014:2},{2002:2},{2007:9},{2012:1},{2000:2},{2005:1},{2010:8}]]
B00001P505 [[{2006:5},{2011:6},{2004:1},{2009:4},{2014:11},{2000:3},{2005:2},{2010:6},{2003:2},{2008:7},{2013:17},{2002:4},{2007:2},{2012:6}]]
B00001W0EQ [[{2002:2},{2007:7},{2012:9},{2008:5},{2013:11},{2004:1},{2009:5},{2014:13},{2000:1},{2005:4},{2010:5},{2001:1},{2006:2},{2011:4}]]
B00001W0DI [[{2002:3},{2007:8},{2012:6},{2006:4},{2011:8},{2003:3},{2008:6},{2013:6},{2004:4},{2009:8},{2014:2},{2000:1},{2005:8},{2010:2}]]
B00004SX4H [[{2003:1},{2008:1},{2013:15},{2000:1},{2005:3},{2010:1},{2001:1},{2006:1},{2011:4},{2009:3},{2014:8},{2002:4},{2007:4},{2012:4}]]
B00001P4ZH [[{2001:1},{2006:16},{2011:60},{2004:4},{2009:44},{2014:62},{2000:2},{2005:17},{2010:57},{2007:14},{2012:64},{2003:4},{2008:27},{2013:110}]]
B0000228GF [[{2001:2},{2006:3},{2011:1},{2002:2},{2012:2},{2000:1},{2005:3},{2010:1},{2003:2},{2008:1},{2013:2},{2004:2},{2009:3},{2014:3}]]
B00004SB92 [[{2002:54},{2007:15},{2012:4},{2000:28},{2005:12},{2010:8},{2004:21},{2009:7},{2014:4},{2001:41},{2006:9},{2011:6},{2003:16},{2008:11},{2013:8}]]
B00001P4XA [[{2002:2},{2007:23},{2012:12},{2001:1},{2006:19},{2011:5},{2004:7},{2009:7},{2014:10},{2003:4},{2008:19},{2013:21},{2000:3},{2005:17},{2010:8}]]
Time taken: 221.238 seconds, Fetched: 63006 row(s)
hive>
```

Sample output screenshot



Business Questions Answered

- Q8. Visualization of results observe
 - Reviews data is a subset of ratings dataset with additional details of review comments, helpful, summary, etc.,.
 - Reviews adoption improved drastically from year 2007-2008.
 - Ratings adoption improved drastically from year 2006-2007.
 - Number of reviewer's comments per year started to multiple from year 2007, but in 2014 there was a drop in number unique reviewers.
 - Number of users per year started to multiple from year 2004, but in 2014 there was drop in number unique users.
 - As a trend number items reviewed each year is directly proportional to the number of users, same trend can be seen even for the drop in number of reviews during year 2014.

Conclusion

- Amazon Electronics Products data loaded and analysed successfully.
- Created an Database in Hive and loaded Reviews/Rating data as hive tables.
- Partitioned the Hive tables to answer the business questions using Hive queries.
- Code: <https://github.com/ssushmanth/AmazonElectronicsAnalysis-Hive>