



Analysis of Amazon Movie and TV Reviews (2005-2015)

Jasmine Diep, William Lam,

Nina Roberts, Taron Sarksyian

CIS 5200 - Professor Jongwook Woo



Introduction

- Amazon Prime offers access to unlimited streaming of thousands of movies and TV episodes
- Analysis of top 100 Amazon movie and TV titles from 2005 to 2015
 - High average rating
 - Scale of 1-5
- Most reviewed movie and TV titles
 - Average star rating
- Top 20 reviews



Amazon Dataset

Analysis of Amazon Movie and TV reviews collected by Amazon and stored on AWS.

Consists of:

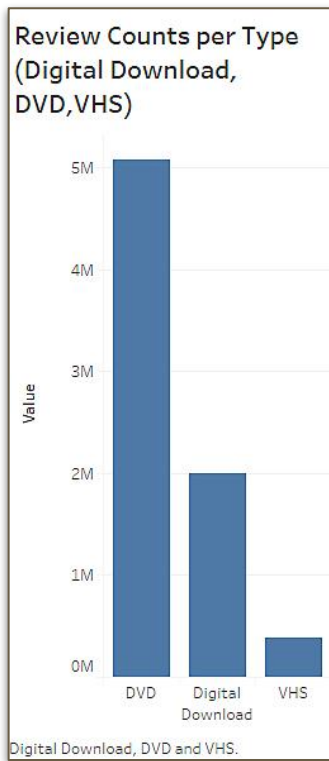
- DVD Reviews
- Digital Download Reviews
- VHS Reviews

Instructions for use found at:

<https://s3.amazonaws.com/amazon-reviews-pds/readme.html>



Amazon Dataset (cont.)



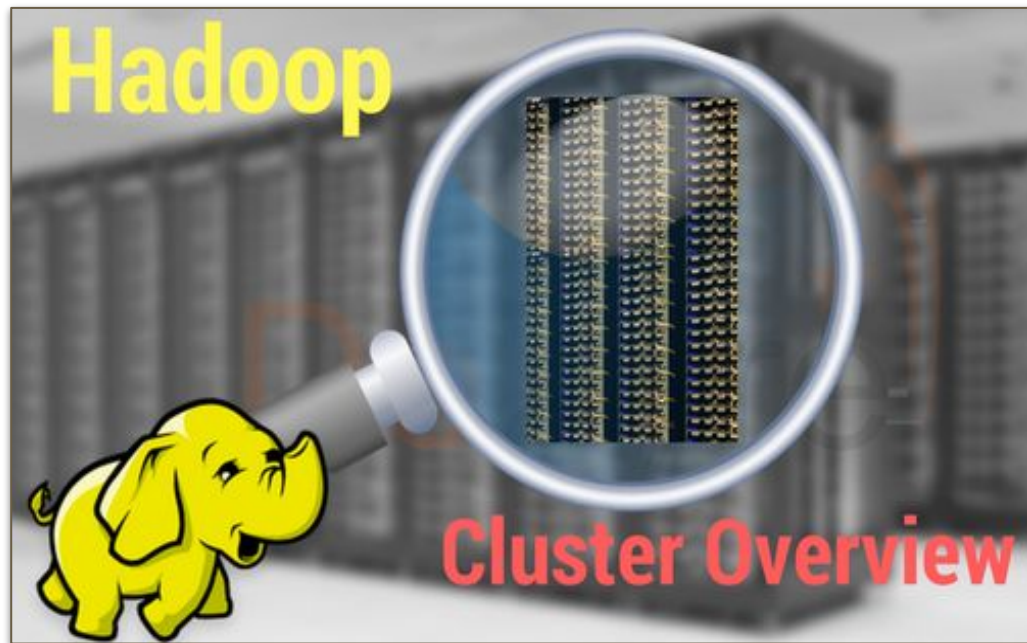
Review Distribution:

- DVD Reviews - *Size: 3.7GB*
 - 5,069,149 Reviews
- Digital Download Review - *Size: 1.2GB*
 - 2,003,323 Reviews
- VHS Reviews - *Size: 337MB*
 - 380,604 Reviews

Total: 5.33GB

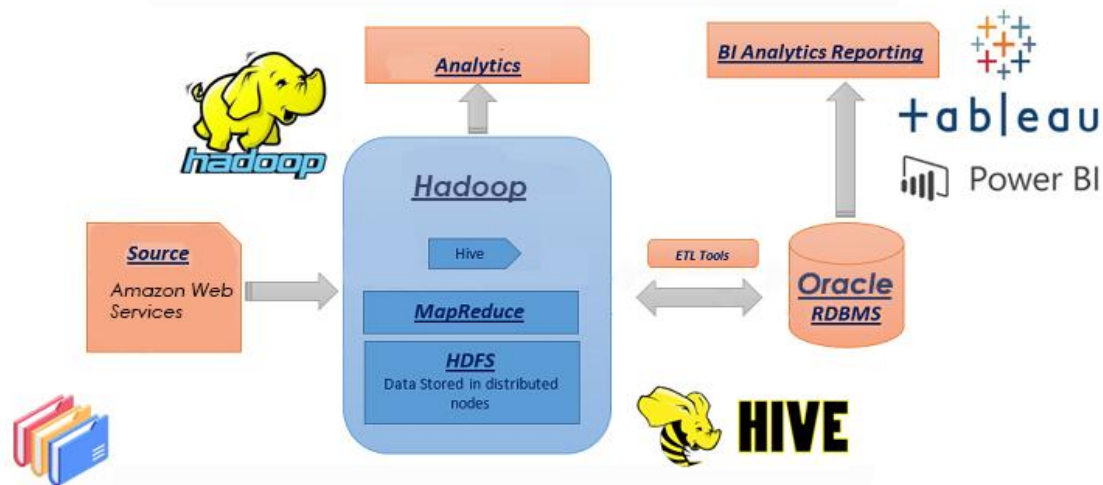
Platform Specs

- Cluster Version: 20.3.3-20
- Cluster # of Nodes: 3
- # of CPUs Cores: 12
- Memory Size: 180GB
- Storage Size: 957GB



Software and Tools

Big Data ETL Architecture on Oracle Cloud



Data files: Amazon Customer Reviews Library

DVD reviews: https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Video_DVD_v1_00.tsv.gz

Streaming Reviews: https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Digital_Video_Download_v1_00.tsv.gz

VHS Reviews: https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Video_v1_00.tsv.gz

Location: AWS

Type: tsv.gz (compressed tsv)



```
1 --CIS 5200 Project Lab
2 --Amazon Movie Review Analysis
3
4 --PART 1
5 --Amazon Movie Review (Streaming & DVD) top 100 for timeframe of review data
6
7 --Get work files from AWS:
8 wget -O amazon_data1.tsv.gz https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Digital_Video_Download
9 wget -O amazon_data2.tsv.gz https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Video_V1_00.tsv.gz
10 wget -O amazon_data3.tsv.gz https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Video_V1_00.tsv.gz
11
12 --Create work directory:
13 hdfs dfs -mkdir amazon_data
14
15 --Unzip and save .tsv files to HDFS:
16 gzip -d amazon_data1.tsv.gz | hdfs dfs -put amazon_data1.tsv amazon_data
17 gzip -d amazon_data2.tsv.gz | hdfs dfs -put amazon_data2.tsv amazon_data
18 gzip -d amazon_data3.tsv.gz | hdfs dfs -put amazon_data3.tsv amazon_data
19
20 --New terminal - connect to HIVE:
21 hdfs dfs -chmod -R o+w .
22 beeline
23 use your database name
24
25 --Create table to store data from two previously unzipped files:
26 CREATE EXTERNAL TABLE IF NOT EXISTS amazon (
27 marketplace string,
28 customer_id string,
29 review_id string,
30 product_id string,
31 product_parent string,
32 product_title string,
33 product_category string,
34 star_rating int,
35 helpful_votes int,
36 total_votes int,
37 view string,
38 verified_purchase string,
39 review_headline string,
40 review_body string,
41 review_date date)
42 ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
43 STORED AS TEXTFILE LOCATION '/user/nrobert/amazon_data'
44 TBLPROPERTIES ('skip.header.line.count'='1');
45
46 --Review table headers:
47 DESCRIBE amazon;
48
49 --Review distinct titles
50 SELECT DISTINCT PRODUCT_ID, PRODUCT_TITLE FROM AMAZON;
51
52 --Count distinct titles
53 SELECT COUNT(DISTINCT PRODUCT_ID) FROM AMAZON;
54
55 --Review review years and counts
56 SELECT EXTRACT(YEAR FROM REVIEW_DATE) AS YEAR FROM AMAZON
57 ORDER BY YEAR
58
59 SELECT EXTRACT(YEAR FROM REVIEW_DATE) AS YEAR,
60 COUNT (*) AS YEAR_COUNT
61 FROM AMAZON
62 GROUP BY EXTRACT(YEAR FROM REVIEW_DATE)
63 ORDER BY YEAR;
64
65 --Run query to select top 100 movie reviews by average star rating
66 SELECT product_title, ROUND(avg(star_rating), 2) as average_rating, COUNT(review_id) as review_count FROM AMAZON
67 WHERE (REVIEW_DATE >= '01-Jan-05' AND REVIEW_DATE <= '31-Dec-15')
68 GROUP BY product_title
69 HAVING avg(star_rating) >= 5
70 ORDER BY review_count DESC, average_rating;
71
72 --Run query to select top 100 movie reviews by average star rating and count of ratings
73 SELECT
74 product_title,
75 ROUND(avg(star_rating), 2) as average_rating,
76 COUNT(review_id) as review_count
77 FROM AMAZON
78 WHERE (review_date >= '01-Jan-05' AND review_date <= '31-Dec-15') and (product_title != 'Pilot')
79 GROUP BY product_title HAVING avg(star_rating) >= 4.5
80 ORDER BY review_count DESC
81 LIMIT 100;
82
83 --Create tmp directory:
84 hdfs dfs -mkdir tmp;
85
86 --Create top 100 table by avg(star_rating) and review count and Save as comma delimited file in HDFS tmp directory:
87 CREATE TABLE IF NOT EXISTS amazon_top
88 ROW FORMAT DELIMITED
89 FIELDS TERMINATED BY ','
90 STORED AS TEXTFILE LOCATION '/user/nrobert/tmp'
91 AS
92 SELECT
93 product_title,
94 ROUND(avg(star_rating), 2) as average_rating,
95 COUNT(review_id) as review_count
96 FROM amazon
97 WHERE (review_date >= '01-Jan-05' AND review_date <= '31-Dec-15') and (product_title != 'Pilot')
98 GROUP BY product_title HAVING avg(star_rating) >= 4.5
99 ORDER BY review_count DESC
100 LIMIT 100;
101
102 --View Top 100 file:
103 hdfs dfs -cat tmp/000000_0
104 hdfs dfs -put tmp/000000_0
105
106 --PART 2
107 --Sentiment Analysis - Trending words Over Time
108
109 --Download dictionary and move to tmp directory on HDFS
110 wget -O dictionary.tsv https://s3.amazonaws.com/nlpcloudatasets/dictionary.tsv
111 hdfs dfs -put dictionary.tsv /user/nrobert/tmp
112
113 --Create table for dictionary
114 CREATE EXTERNAL TABLE dictionary (
115 type string,
116 length int,
117 word string,
118 pos string,
119 stemmed string,
120 polarity string)
121 ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
122
123 LOAD DATA INPATH '/user/nrobert/tmp/dictionary.tsv' OVERWRITE INTO TABLE dictionary;
124
125 --Create Views for sentiment analysis and keyword popularity
126 CREATE VIEW reviews_simple AS
127 SELECT
128 review_id,
129 review_body,
130 review_raw;
131
132 FROM reviews_raw;
133
134 CREATE VIEW reviews_clean AS
135 SELECT
136 review_id,
137 review_body
138 FROM reviews_simple ;
139
140 create view ll as select review_id, words from reviews_raw lateral view explode(sentences(lower(review_body))) dummy as words;
141
142 create view ll2 as select review_id, word from ll lateral view explode(words) dummy as word ;
143
144 create view ll3 as select
145 review_id,
146 ll2.word,
147 case d.polarity
148 when 'negative' then -1
149 when 'positive' then 1
150 else 0 end as polarity
151 from ll2 left outer join dictionary d on ll2.word = d.word;
152
153 create table reviews_sentiment stored as orc as select
154 review_id,
155 case
156 when sum(polarity) > 0 then 'positive'
157 when sum(polarity) < 0 then 'negative'
158 else 'neutral' end as sentiment
159 from ll3 group by review_id;
160
161 CREATE TABLE reviews1
162 STORED AS ORC
163 AS
164 SELECT
165 t.*,
166 case s.sentiment
167 when 'positive' then 1
168 when 'neutral' then 0
169 when 'negative' then -1
170 end as sentiment
171 FROM reviews_clean t LEFT OUTER JOIN reviews_sentiment s on t.id = s.id;
172
173 CREATE VIEW reviews_trending_words AS
174 SELECT
175 review_id,
176 review_body,
177 review_raw;
178
179 create table reviews_trending_words_struct ((ID,TITLE ARRAY<STRUCT<(ngram array<string>, estfrequency double)>>));
180
181 INSERT OVERWRITE TABLE reviews_trending_words_struct SELECT context_ngrams(sentences(lower(text)), array(null), 10) AS snippets from reviews_raw;
182
183 create table reviews_trending_words (ngram string, estfrequency double);
184
185 INSERT OVERWRITE TABLE reviews_trending_words select X.ngram[0], X.estfrequency from (select explode(review_text) as X from reviews_trending_words_struct ) Z;
```

- Github Project Link: https://github.com/NinaRo2/CIS_5200
- Total: 182 Lines of Code

Data Analysis Methodology

- Star Rating (1-5 scale)
- Average Star Rating
- Product Title (Movies and TV)
- Review Date
- Review Count

Frozen > Customer reviews

Customer reviews

★★★★☆ 4.7 out of 5

56,050 global ratings



Frozen
by Kristen Bell

5 star	<div><div></div></div>	85%
4 star	<div><div></div></div>	8%
3 star	<div><div></div></div>	4%
2 star	<div><div></div></div>	1%
1 star	<div><div></div></div>	2%

[Write a review](#)

[How are ratings calculated?](#)

Top positive review

[All positive reviews >](#)

 MikeC

★★★★★ **Precious family film!**

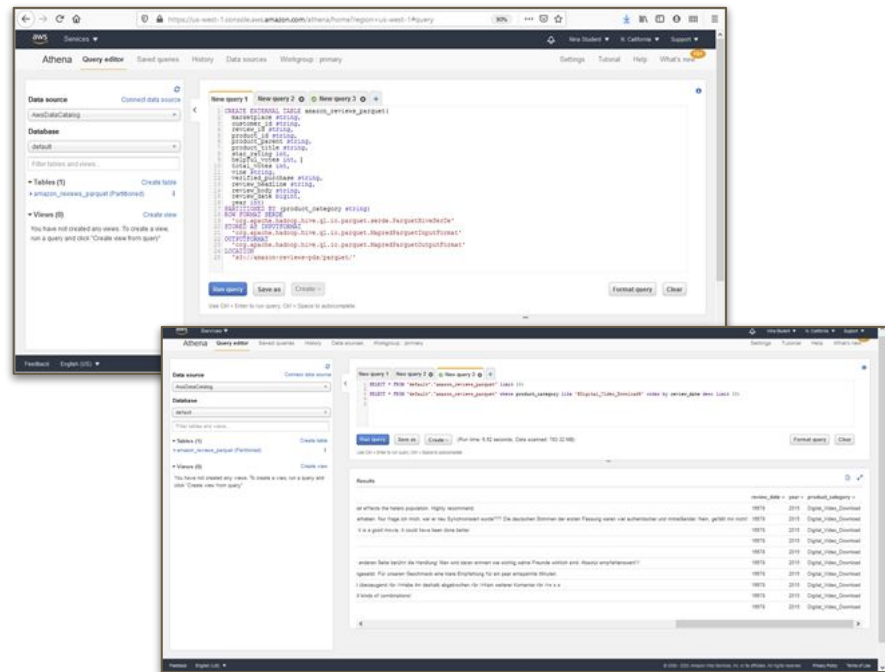
Reviewed in the United States on May 20, 2018

2 yr. old nephew LOOOVES this one. This is actually the first feature length film that held his attention for almost the entire movie, and is the first movie he began to ask for. He knows all of the songs and enjoys humming along. Olaf is his favorite character, and he also loves "Let It Go" and will do a dance along with the song. Too adorable. The movie is sweet, beautifully illustrated, and jam packed with extremely catchy tunes. Idina Menzel and Kristen Bell are amazingly talented in their vocal range and ability, and make for a movie that IS for children, but can be enjoyed by adults as well. This one is worth watching!

41 people found this helpful

Implementation

- Verified no new data existed in reviews db on AWS (2016 - 2020)
- Downloaded (wget) .gz files to local drive on class server
- Unzipped files (gzip)
- Created work directory and moved files to HDFS



Implementation (cont.)

- Three files were then merged using the '-getmerge' file command
- Total size of raw data file = 5.33GB
- Used 'cat' command to view data
- Multiple ways to combine the data from multiple files including joining tables in Pig

```
-bash-4.1$ hdfs dfs -getmerge /user/nrobert/amazon_data/ /home/nrobert/combined.tsv
-bash-4.1$ ls -al
total 5250592
drwx----- 2 nrobert nrobert      4096 Nov 16 21:54 .
drwxr-xr-x 40 root    root        4096 Nov 11 22:13 ..
-rw----- 1 nrobert nrobert       786 Nov 16 21:26 .bash_history
-rw-r--r-- 1 nrobert nrobert 5334908916 Nov 16 21:55 combined.tsv
-rw-r--r-- 1 nrobert nrobert 41678984 Nov 16 21:55 .combined.tsv.crc
-bash-4.1$ cat combined.tsv | head -20
marketplace customer_id review_id product_id product_parent product_title product_category star_rating helpful
votes total_votes vine verified_purchase review_headline review_body review_date
US 12190288 R3FU16928EP5TC B00AYB1482 668895143 Enlightened: Season 1 Digital_Video_Download 5 0 0 N
Y I loved it and I wish there was a season 3 I loved it and I wish there was a season 3... I watched season 2 and loved that as well!
```

Implementation (cont.)

Table Creation in HIVE

- Used CREATE TABLE command to create and load data to table
- Used ANALYZE TABLE function

ANALYZE TABLE amazon COMPUTE STATISTICS;

*[Table **nrobert.amazon** stats: [numFiles=1, numRows=9506893, totalSize=5334908916, rawDataSize=5325401831]*

- Computed 407,787 distinct titles via query

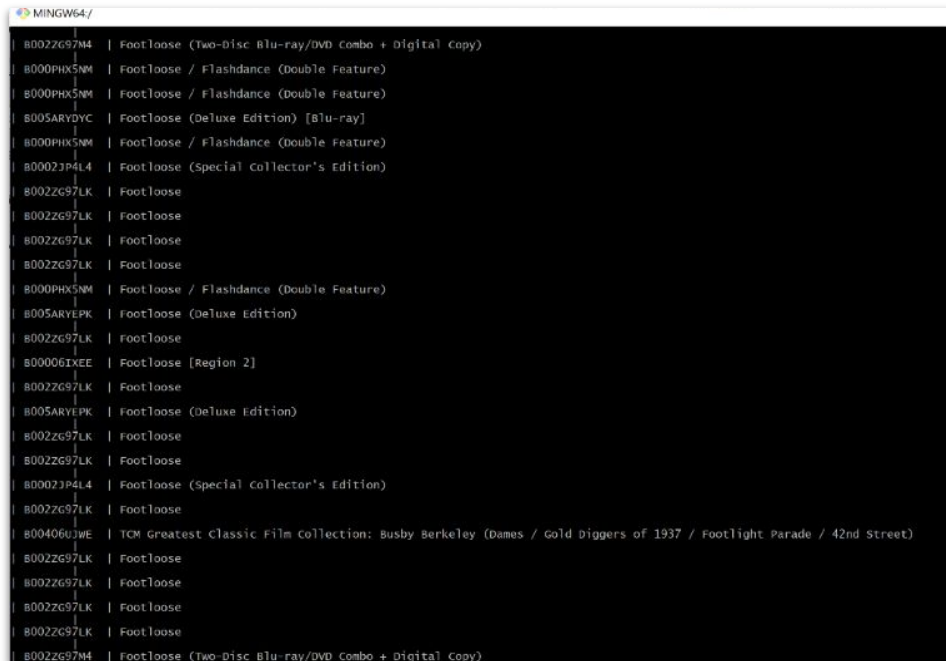
```
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd> review_headline string,  
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd> review_body string,  
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd> review_date date  
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd> );  
No rows affected (0.226 seconds)  
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd> show tables;  
+-----+  
| tab_name |  
+-----+  
| amazon   |  
+-----+  
1 row selected (0.189 seconds)  
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd> describe amazon;  
+-----+-----+-----+  
| col_name | data_type | comment |  
+-----+-----+-----+  
| marketplace | string | |  
| customer_id | string | |  
| review_id | string | |  
| product_id | string | |  
| product_parent | string | |  
| product_title | string | |  
| product_category | string | |  
| star_rating | int | |  
| helpful_votes | int | |  
| total_votes | int | |  
| vine | string | |  
| verified_purchase | string | |  
| review_headline | string | |  
| review_body | string | |  
| review_date | date | |  
+-----+-----+-----+  
15 rows selected (0.213 seconds)  
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd>
```

Data Normalization in HIVE

- Grouping by Decision
 - Movie and TV titles had multiple product_ids
 - Product_titles were amended with various descriptive phrases

i.e. “Footloose (Double Feature)”
and “Footloose (Deluxe Edition)”

- Solution: removed phrases to normalize the data



```
MINGW64/  
| B0022G97M4 | Footloose (Two-Disc Blu-ray/DVD Combo + Digital Copy)  
| B000PHX5NM | Footloose / Flashdance (Double Feature)  
| B000PHX5NM | Footloose / Flashdance (Double Feature)  
| B005ARYDYC | Footloose (Deluxe Edition) [Blu-ray]  
| B000PHX5NM | Footloose / Flashdance (Double Feature)  
| B0002JP4L4 | Footloose (Special Collector's Edition)  
| B0022G97LK | Footloose  
| B0022G97LK | Footloose  
| B0022G97LK | Footloose  
| B000PHX5NM | Footloose / Flashdance (Double Feature)  
| B005ARYEPK | Footloose (Deluxe Edition)  
| B0022G97LK | Footloose  
| B00006IXEE | Footloose [Region 2]  
| B0022G97LK | Footloose  
| B005ARYEPK | Footloose (Deluxe Edition)  
| B0022G97LK | Footloose  
| B0022G97LK | Footloose  
| B0002JP4L4 | Footloose (Special Collector's Edition)  
| B0022G97LK | Footloose  
| B00406UJWE | TCM Greatest Classic Film Collection: Busby Berkeley (Dames / Gold Diggers of 1937 / Footlight Parade / 42nd Street)  
| B0022G97LK | Footloose  
| B0022G97LK | Footloose  
| B0022G97LK | Footloose  
| B0022G97LK | Footloose  
| B0022G97M4 | Footloose (Two-Disc Blu-ray/DVD Combo + Digital Copy)
```

Data Normalization in HIVE (cont.)

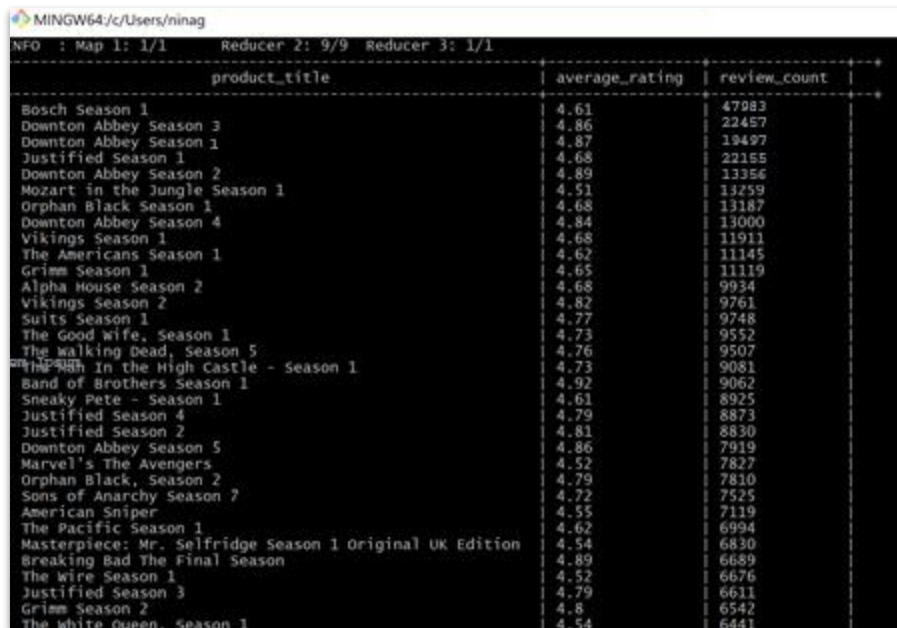
- For movies between 1990 and 2011, movie year was saved in the product_title field
- The following prevented accurate grouping
 - Product_title contains both "A" and "a"
 - Product_title contains both "The" and "the"
 - Product_title contains both "Its" and "It's"
- Solution: regexp_replace command
 - `SELECT product_title = regexp_replace (product_title,'s*\[^\]^*\','") FROM amazon;`
 - `SELECT product_title = regexp_replace (product_title,'s*\[^\]^*\','") FROM amazon;`

Note: This caused some titles to be erroneously grouped together. They were identified and manually fixed. I.e. Midway, Little Women

Top 100 Reviews

The following query was used to query the Top 100 Reviews:

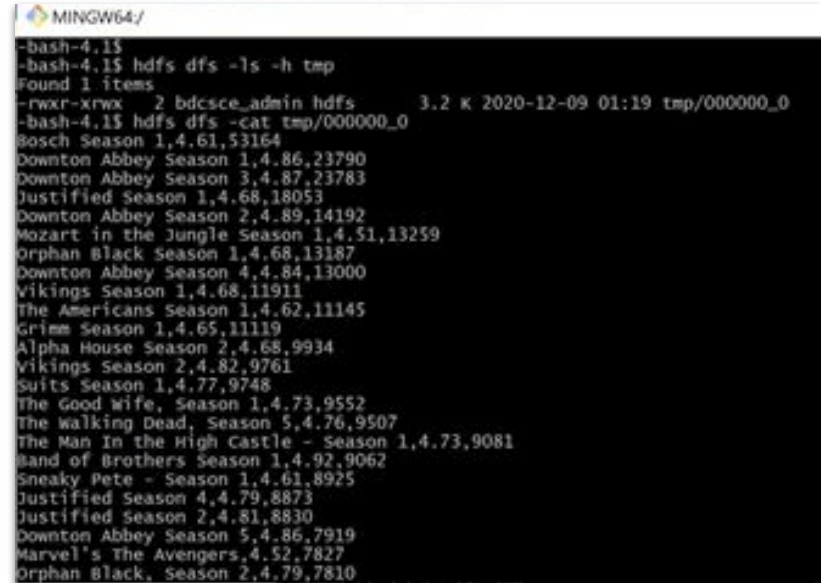
```
SELECT  
product_title,  
ROUND(avg(star_rating), 2) as average_rating,  
COUNT(review_id) as review_count  
FROM amazon  
WHERE (review_date >= '01-Jan-05' AND  
review_date <= '31-Dec-15')  
and (product_title != 'Pilot')  
GROUP by product_title HAVING avg(star_rating) >= 4.5  
ORDER by review_count DESC  
LIMIT 100;
```



product_title	average_rating	review_count
Bosch Season 1	4.61	47983
Downton Abbey Season 3	4.86	22457
Downton Abbey Season 1	4.87	19497
Justified Season 1	4.68	22155
Downton Abbey Season 2	4.89	13356
Mozart in the Jungle Season 1	4.51	13259
Orphan Black Season 1	4.68	13187
Downton Abbey Season 4	4.84	13000
Vikings Season 1	4.68	11911
The Americans Season 1	4.62	11145
Grimm Season 1	4.65	11119
Alpha House Season 2	4.68	9934
Vikings Season 2	4.82	9761
Suits Season 1	4.77	9748
The Good wife, Season 1	4.73	9552
The Walking Dead, Season 5	4.76	9507
The Man in the High Castle - Season 1	4.73	9081
Band of Brothers Season 1	4.92	9062
Sneaky Pete - Season 1	4.61	8925
Justified Season 4	4.79	8873
Justified Season 2	4.81	8830
Downton Abbey Season 5	4.86	7919
Marvel's The Avengers	4.52	7827
Orphan Black, Season 2	4.79	7810
Sons of Anarchy Season 7	4.72	7525
American Sniper	4.55	7119
The Pacific Season 1	4.62	6994
Masterpiece: Mr. Selfridge Season 1 Original UK Edition	4.54	6830
Breaking Bad The Final Season	4.89	6689
The Wire Season 1	4.52	6676
Justified Season 3	4.79	6611
Grimm Season 2	4.8	6542
The White Queen Season 1	4.54	6441

Build Data Output File and Export

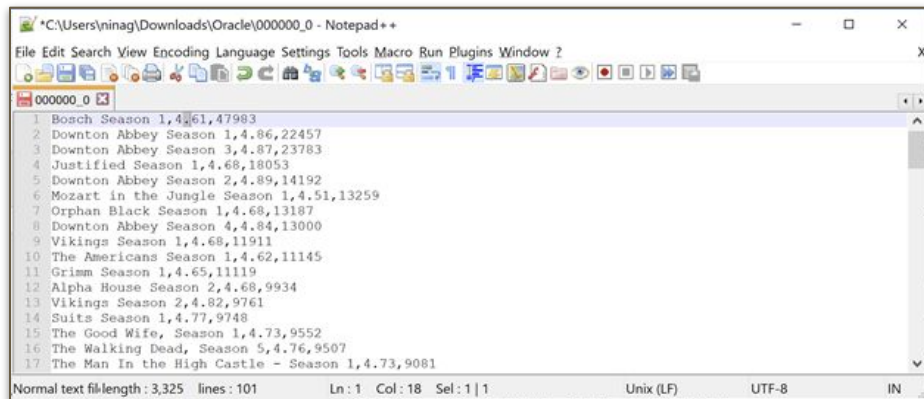
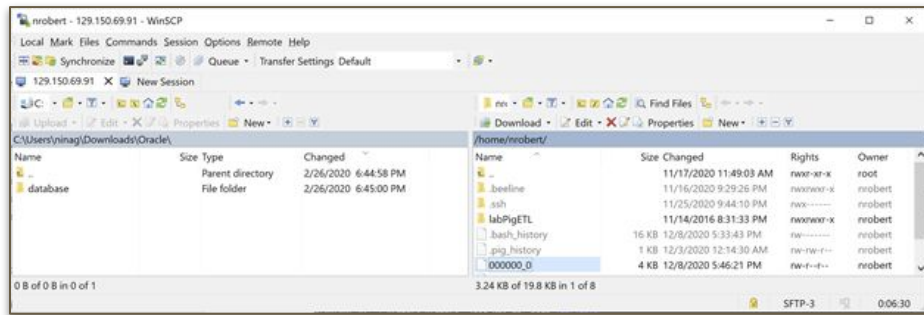
```
CREATE TABLE IF NOT EXISTS amazon_top
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ","
STORED AS TEXTFILE LOCATION "/user/nrobert/tmp"
AS
SELECT
product_title,
ROUND(avg(star_rating), 2) as average_rating,
COUNT(review_id) as review_count
FROM amazon
WHERE (review_date >= '01-Jan-05' AND
review_date <= '31-Dec-15')
and (product_title != 'Pilot')
GROUP by product_title HAVING avg(star_rating) >= 4.5
ORDER by review_count DESC
LIMIT 100;
```



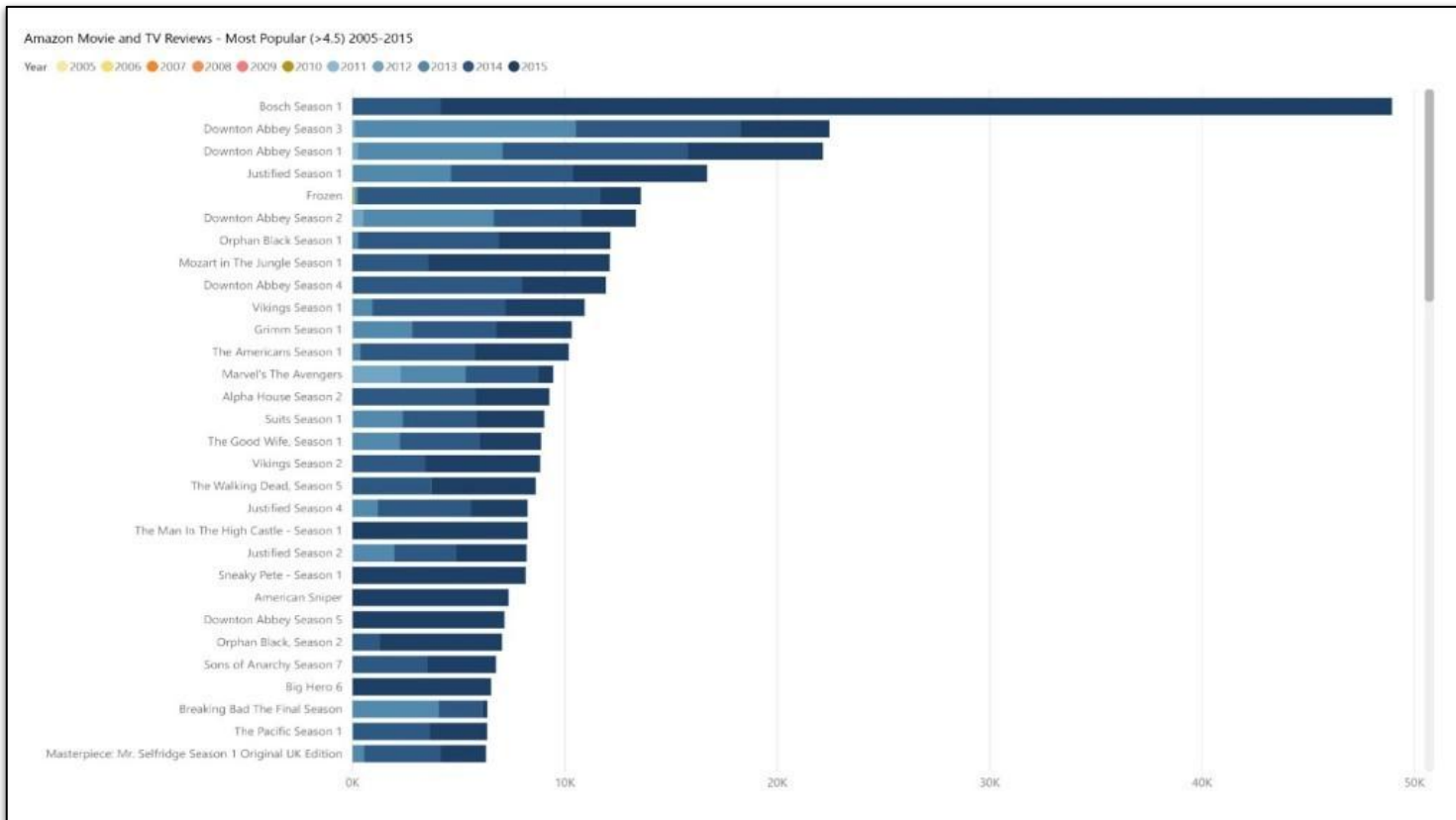
```
MINGW64/
-bash-4.1$
-bash-4.1$ hdfs dfs -ls -h tmp
Found 1 items
-rwxr-xrwx  2 bdcscs_admin hdfs      3.2 K 2020-12-09 01:19 tmp/000000_0
-bash-4.1$ hdfs dfs -cat tmp/000000_0
Bosch Season 1,4.61,53164
Downton Abbey Season 1,4.86,23790
Downton Abbey Season 3,4.87,23783
Justified Season 1,4.68,18053
Downton Abbey Season 2,4.89,14192
Mozart in the Jungle Season 1,4.51,13259
Orphan Black Season 1,4.68,13187
Downton Abbey Season 4,4.84,13000
Vikings Season 1,4.68,11911
The Americans Season 1,4.62,11145
Grimm Season 1,4.65,11119
Alpha House Season 2,4.68,9934
Vikings Season 2,4.82,9761
Suits Season 1,4.77,9748
The Good Wife, Season 1,4.73,9552
The Walking Dead, Season 5,4.76,9507
The Man In the High Castle - Season 1,4.73,9081
Band of Brothers Season 1,4.92,9062
Sneaky Pete - Season 1,4.61,8925
Justified Season 4,4.79,8873
Justified Season 2,4.81,8830
Downton Abbey Season 5,4.86,7919
Marvel's The Avengers,4.52,7827
Orphan Black, Season 2,4.79,7810
```


Build Working Data Output File and Export (cont.)

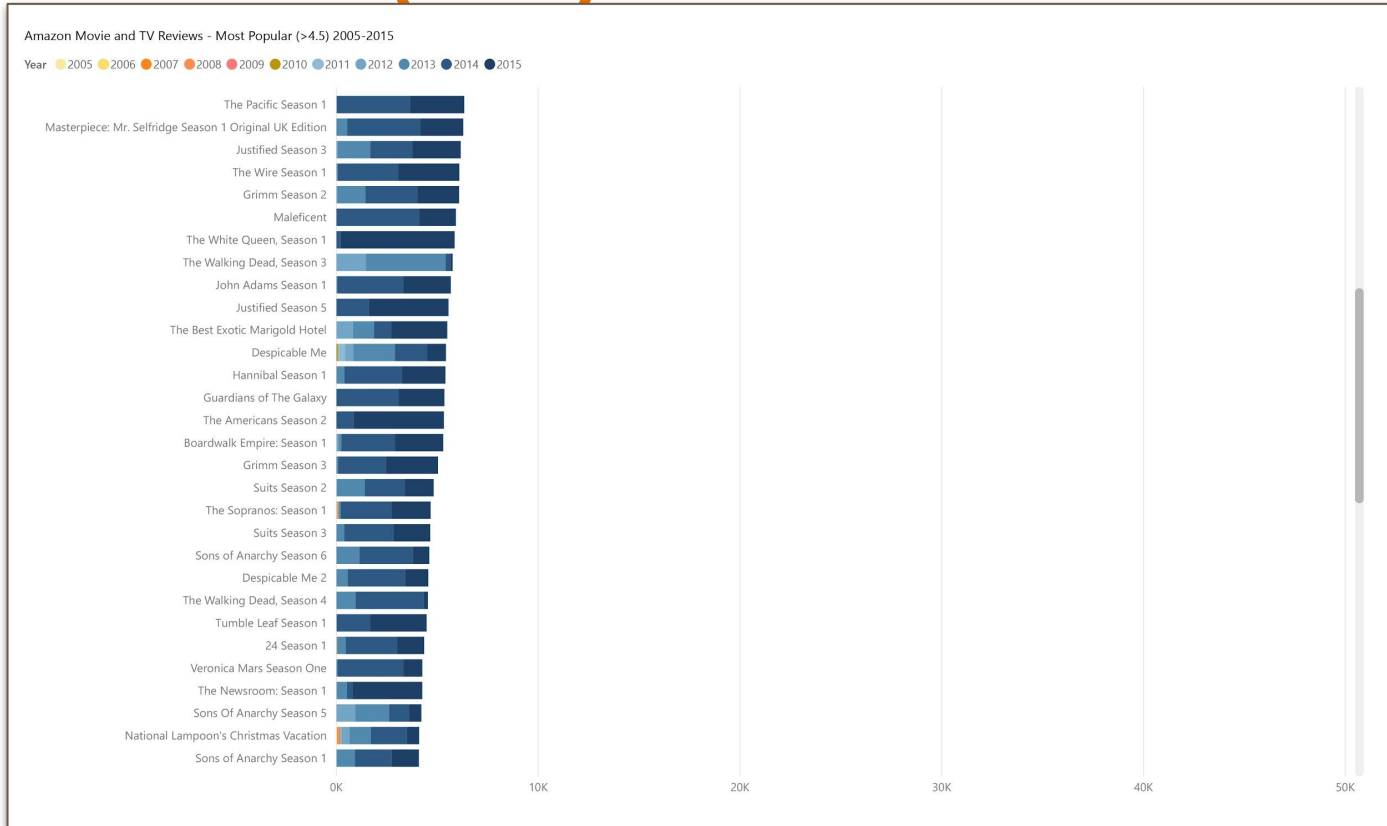
- Move file to local drive
`hdfs dfs -get tmp/000000_0`
- Transfer file to your computer using WinSCP
- View or edit file on computer



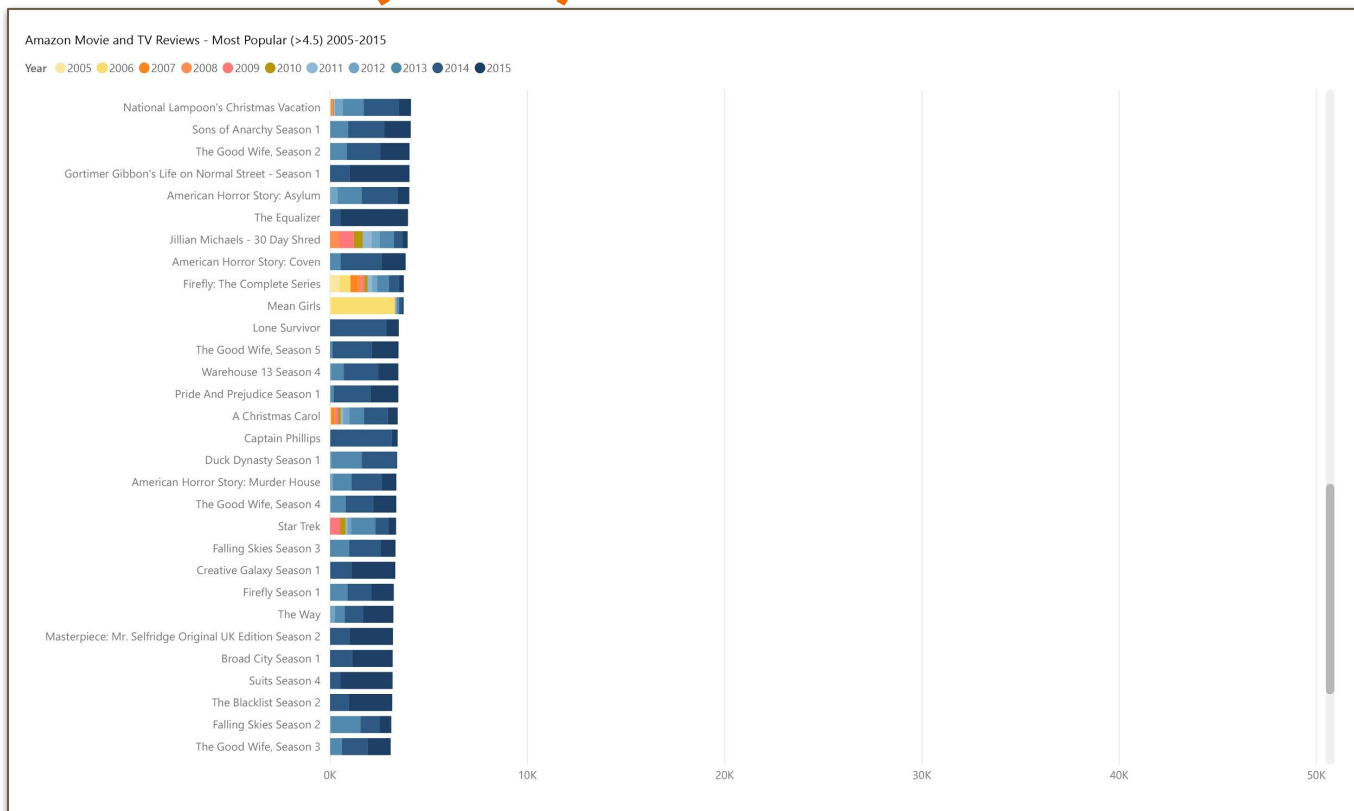
Top 100 Reviews



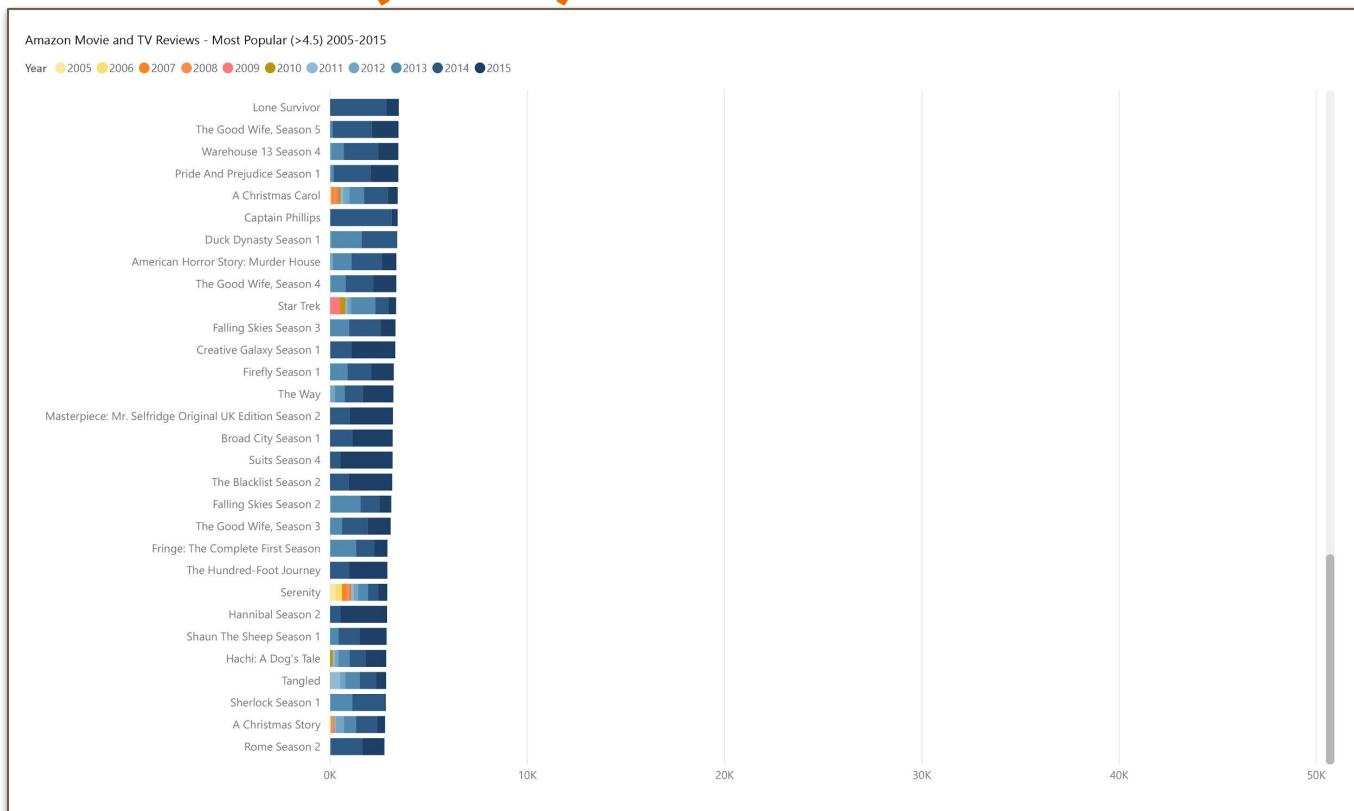
Top 100 Reviews (cont.)



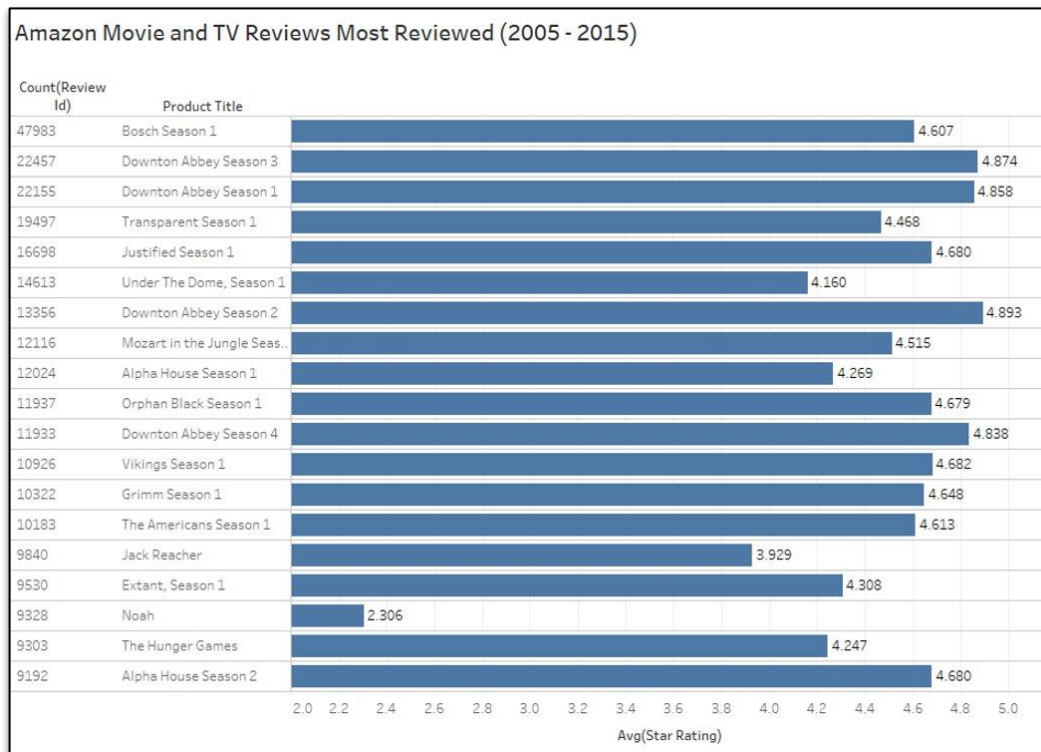
Top 100 Reviews (cont.)



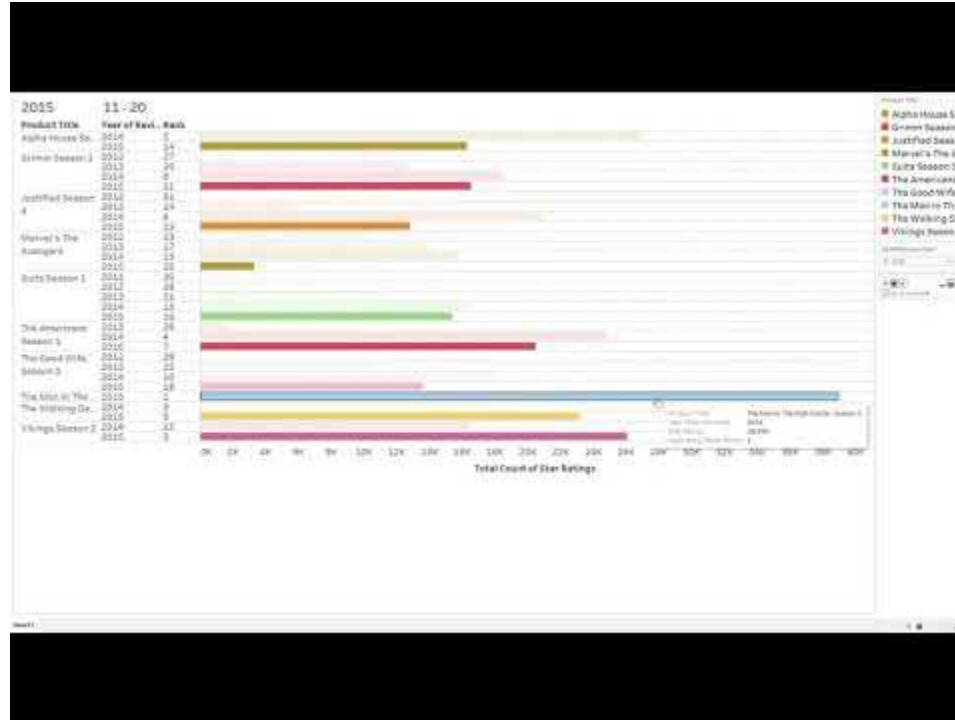
Top 100 Reviews (cont.)



Most Reviewed Movie and TV Titles



Animated Data Visualization (Tableau)



Quality Assurance

- Queried titles from Amazon.com's "Best Sellers in Movie and TV" listing from 2020
- Validated some of our popular titles from 2005 to 2015 were present
 - Mean Girls
 - Star Trek
 - National Lampoon's Christmas Vacation
 - A Christmas Carol
 - A Christmas Story



Q&A

