Amazon Reviews

Ann Eitrheim, Alisa Babikova, & YingKun Zhu
March 13, 2019

Overview

Project Objective

Data Used

Infrastructure Utilized

Problems Encountered

Use Cases 1 and 2

Machine Learning Models

Project objective

Using the text from an item's review, predict the overall rating a reviewer gave and how many helpful votes the review will get.

Data Overview

Our Amazon Review dataset contains product reviews including ratings, review text, and helpfulness rating of the review from May 1996 - July 2014.

The dataset that contains the item information includes data such as title, description, image URL, brand info, price, sales price, and purchasing links.

Amazon Reviews

58GB

82,677,139 rows

Model-ready dataframe has 12,759,707 records

7,261,083 unique reviewers

Item Information

11GB

9,430,088 rows

Tota Size

69GB

19 columns, 9 of which has more columns nested within

The datasets we used were in a JSON format and had a nested structure.

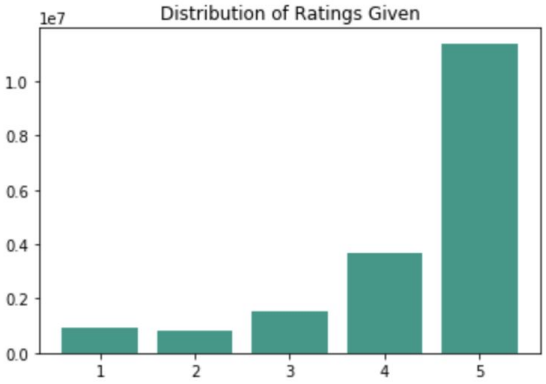

```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the piano.  He is
having a wonderful time playing these old hymns.  The music  is at times
hard to read because we think the book was published for singing from more
than playing from.  Great purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```

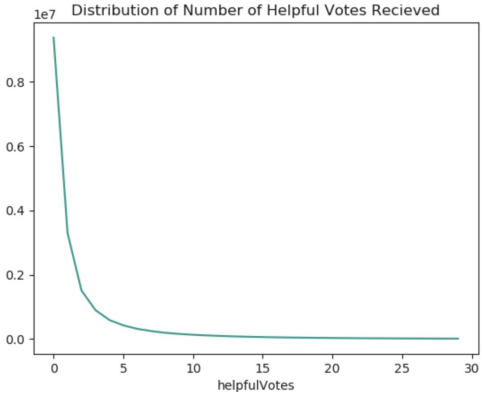

```
metadata.printSchema()
```

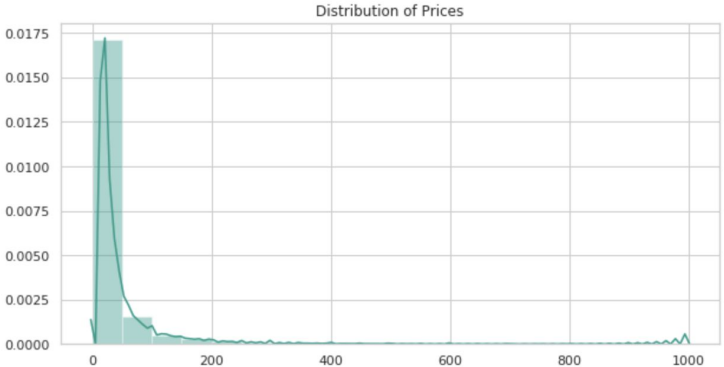
```
root
 |-- _corrupt_record: string (nullable = true)
 |-- asin: string (nullable = true)
 |-- brand: string (nullable = true)
 |-- categories: array (nullable = true)
 |    |-- element: array (containsNull = true)
 |    |    |-- element: string (containsNull = true)
 |-- description: string (nullable = true)
 |-- imUrl: string (nullable = true)
 |-- price: double (nullable = true)
 |-- related: struct (nullable = true)
 |    |-- also_bought: array (nullable = true)
 |    |    |-- element: string (containsNull = true)
 |    |-- also_viewed: array (nullable = true)
 |    |    |-- element: string (containsNull = true)
 |    |-- bought_together: array (nullable = true)
 |    |    |-- element: string (containsNull = true)
 |    |-- buy_after_viewing: array (nullable = true)
 |    |    |-- element: string (containsNull = true)
 |-- salesRank: struct (nullable = true)
 |    |-- Appliances: long (nullable = true)
 |    |-- Arts, Crafts & Sewing: long (nullable = true)
 |    |-- Automotive: long (nullable = true)
 |    |-- Baby: long (nullable = true)
 |    |-- Beauty: long (nullable = true)
 |    |-- Books: long (nullable = true)
 |    |-- Camera &amp; Photo: long (nullable = true)
 |    |-- Cell Phones & Accessories: long (nullable = true)
 |    |-- Clothing: long (nullable = true)
 |    |-- Computers & Accessories: long (nullable = true)
 |    |-- Electronics: long (nullable = true)
 |    |-- Gift Cards Store: long (nullable = true)
 |    |-- Grocery & Gourmet Food: long (nullable = true)
 |    |-- Health & Personal Care: long (nullable = true)
 |    |-- Home &amp; Kitchen: long (nullable = true)
 |    |-- Home Improvement: long (nullable = true)
 |    |-- Industrial & Scientific: long (nullable = true)
 |    |-- Jewelry: long (nullable = true)
 |    |-- Kitchen & Dining: long (nullable = true)
 |    |-- Magazines: long (nullable = true)
```

Descriptive Statistics

summary	overall	totalVotes	helpfulVotes
count	18289296	18289296	18289296
mean	4.30385767718998	4.224146079761627	3.008134430106003
stddev	1.1112010401159536	20.345594293576706	17.2911368753149
min	1	0	0
max	5	24212	23311







Descriptive Statistics

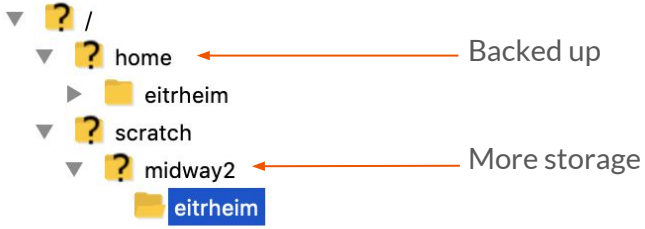
Most reviewed items

asin	title	aveRating	count
439023483	The Hunger Games (The Hunger Games, Book 1)	4.64440601	21398
439023513	Mockingjay (The Final Book of The Hunger Games)	4.23267677	14114
385537859	Inferno	3.91898558	12973
7444117	Allegiant (Divergent, 3)	3.35054240	12629
375831002	The Book Thief	4.62373717	12571

Categories with the most items with reviews

mainCategory	count
Books	12431819
Movies & TV	292552
Toys & Games	18180
CDs & Vinyl	7206
Office Products	2474

Infrastructure Overview



RCC resources: Uploaded data to /scratch/midway2/eitrheim/ as there is much bigger storage quota (but it is not backed up). Our data was too large to fit in /home/eitrheim/.

PySpark for data processing and machine learning.

Problems Encountered and Solved

Handling Hierarchical Schema

asin	helpful	overall	reviewText	reviewTime	reviewerID	summary	reviewID
78	[1, 1]	5	Conversations wit...	2004-08-11	A3AF8FFZAZYNE5	Impactful!	0
116	[5, 5]	4	Interesting Grish...	2002-04-27	AH2L9G3DQHHAJ	Show me the money!	1
116	[0, 0]	1	The thumbnail is ...	2014-03-24	A2IIIDRK3PRRZY	Listing is all sc...	2
868	[10, 10]	4	I'll be honest. I...	2002-09-1			
13714	[0, 0]	4	It had all the so...	2013-10-3			
13714	[0, 0]	5	We have many of t...	2013-07-2			
13714	[0, 0]	5	I love this book....	2014-03-0			
13714	[0, 0]	4	We use this type ...	2013-12-0			
13714	[0, 0]	4	Heavenly Highway ...	2012-10-1			
13714	[2, 3]	5	I bought this for...	2009-09-1			

reviewID	asin	overall	reviewText	reviewTime	reviewerID	summary	totalVotes	helpfulVotes
0	78	5	Conversations wit...	2004-08-11	A3AF8FFZAZYNE5	Impactful!	1	1
1	116	4	Interesting Grish...	2002-04-27	AH2L9G3DQHHAJ	Show me the money!	5	5
2	116	1	The thumbnail is ...	2014-03-24	A2IIIDRK3PRRZY	Listing is all sc...	0	0
3	868	4	I'll be honest. I...	2002-09-11	A1TADCM7YWPQ8M	Not a Bad Transla...	10	10
4	13714	4	It had all the so...	2013-10-31	AWGH7V0BDOJKB	Not the large print	0	0
5	13714	5	We have many of t...	2013-07-27	A3UTQPQPM4TQO0	I was disappointe...	0	0
6	13714	5	I love this book....	2014-03-01	A8ZS0I5L5V31B	GREAT HYMN BOOK!	0	0
7	13714	4	We use this type ...	2013-12-03	ACNGUPJ3A3TM9	Nice Hymnal	0	0
8	13714	4	Heavenly Highway ...	2012-10-16	A3BED5QFJWK88M	Heavenly Highway ...	0	0
9	13714	5	I bought this for...	2009-09-13	A2SUAM1J3GNN3B	Heavenly Highway ...	3	2

Useful function used to flatten the nested structure and access the array within the columns:

```
amazon.select('helpful',F.posexplode("helpful"))
```

This returns a new row for each element in the array with its position

Giving Others Access to Personal RCC Folder

Giving others access to your personal scratch/midway2 folder. Results in only one individual needed to download the large files and lard them to RCC.

hdfs dfs -chmod -R a+rX /user/$USER/data/

That would allow everybody who has an account on RCC to view the files under that directory.

Viral Reviews Skew the Data

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna nec. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna nec. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

The Beach Behemoth Giant Inflatable 12-Foot Pole-to-Pole Beach Ball by Sol Coastal

by Sol Coastal

69 customer reviews | 30 answered questions

2 Price Changes | Add to Droplist

Price: $95.96 & **FREE Shipping**. Details

Get $40 off instantly: Pay $55.96 upon approval for the Amazon.com Store Card.

prime | Try Fast, Free Shipping

- GIGANTIC: Measuring 12 feet from pole to pole, we're pretty sure this puts other "jumbo" beach balls to shame
- CLASSIC: With an astounding 659,000 cubic inch volume, it's just like a classic beach ball, only gianter
- DURABLE: Made of thick, durable 30mil vinyl with reinforced seams, so play hard!
- FULL OF HOT AIR: Large, secure airtight valve keeps air in and won't tear or detach from the vinyl
- DON'T BLOW IT: Electric pumps only! Please do not manually inflate. We had an intern try it, and he's still blowing

Reid hamlin

★★☆☆☆ **A fun way to ruin a weekend and blow 100 bucks.**

February 3, 2018

We took this ball to the beach and after close to 2 hours to pump it up, we pushed it around for about 10 fun filled minutes. That was when the wind picked it up and sent it huddling down the beach at about 40 knots. It destroyed everything in its path. Children screamed in terror at the giant inflatable monster that crushed their sand castles. Grown men were knocked down trying to save their families. The faster we chased it, the faster it rolled. It was like it was mocking us. Eventually, we had to stop running after it because its path of injury and destruction was going to cost us a fortune in legal fees. Rumor has it that it can still be seen stalking innocent families on the Florida panhandle. We lost it in South Carolina, so there is something to be said about its durability.

10,738 people found this helpful

Uranium Ore

by Images SI

Patrick J. McGovern

★★★☆☆ **Great Product, Poor Packaging**

May 14, 2009

I purchased this product 4.47 Billion Years ago and when I opened it today, it was half empty.

27,239 people found this helpful

Improvements to Design and Execution

1. Reduced dataset using random sampling.
2. Clearing the cache and removing RDDs once done using them.
3. Running commands such as: .toPandas(), .show(), and .count() only when needed.
4. Set up the spark session with needed configuration, and switch to kernel 4G 32e.

```
#Manually remove these RDDs instead of waiting for it to fall out of the cache
df.unpersist()
amazon.unpersist()
metadata.unpersist()
df1.unpersist()
df2.unpersist()

from pyspark.sql import SparkSession
spark.catalog.clearCache()

from pyspark.sql import SQLContext
sqlContext.clearCache()
```

```
#change configuration settings on Spark
conf = spark.sparkContext._conf.setAll([('spark.executor.memory', '256g'),
                                        ('spark.app.name', 'Spark Updated Conf'),
                                        ('spark.executor.cores', '16'),
                                        ('spark.cores.max', '16'),
                                        ('spark.driver.memory','256g'),
                                        ("spark.sql.autoBroadcastJoinThreshold", -1),
                                        ("mapreduce.map.memory.mb", -1),
                                        ("mapreduce.reduce.memory.mb", -1),
                                        ("spark.yarn.executor.memoryOverhead", -1)])
```

Preparing the dataset for Modeling

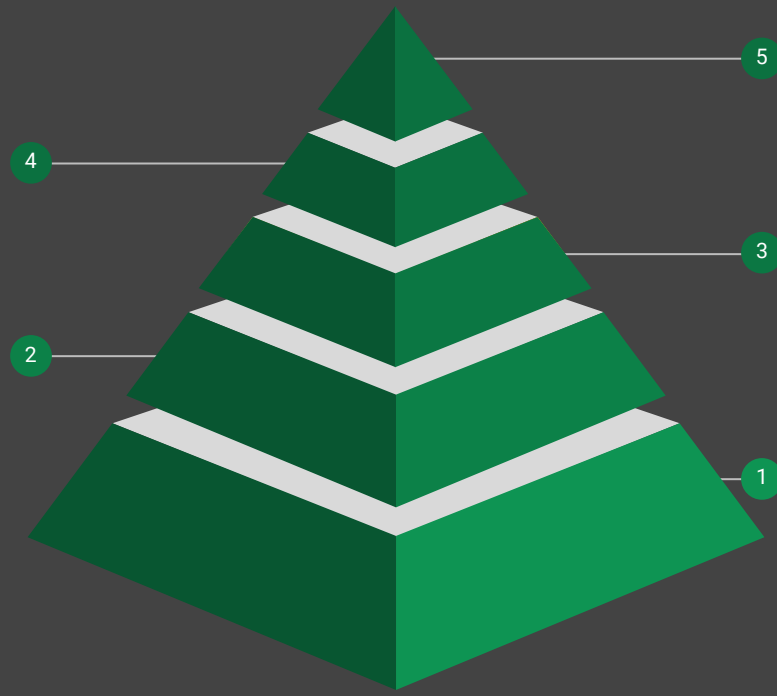
helpfulVotes -> high medium low

Number of helpful votes a single review can receive varies. Bucket into three bins.

Take a sample from the df

The combined dataset has 12,762,877 records over 11 columns.

df.sample(False, 0.1, 43)
#replacement, fraction, seed



reviewText -> NLP steps

A more detailed process flow chart on the next slide.

Overall rating -> binary

Positive <- 1 #for larger and equal to 4 out 5;
Else <- 0 #for below 4 out of 5.

Merge item with metadata

And get rid of the hierarchy structure embedded in the metadata

Buckets (0,1) for positive or negative review

Buckets (1,2,3) for number of helpful votes

reviewText	label	overall	helpfulVotes	hvcat
This one novel, m...	1	5	1	2.0
As usual, excelle...	1	5	0	1.0
The Portable Geni...	1	5	7	2.0
Amidst the overla...	1	5	3	2.0
The title suggest...	0	2	2	2.0

label	overall	helpfulVotes	hvcat	words	filtered
1.0	3	4	2.0	[but, still, ente...	[still, entertain...
1.0	5	0	1.0	[cannot, put, the...	[put, book, down....
0.0	2	0	1.0	[excluding, the, ...	[excluding, langu...
1.0	4	1	2.0	[first, of, all, ...	[first, warning, ...
1.0	4	1	2.0	[imaginative, and...	[imaginative, dar...

label	overall	helpfulVotes	hvcat	words	filtered	rawFeatures	features
1.0	3	1	2.0	[the, real, quest...	[real, question, ...	(20,[0,1,2,3,4,5,...	(20,[0,1,2,3,4,5,...
1.0	3	4	2.0	[but, still, ente...	[still, entertain...	(20,[0,1,2,3,4,5,...	(20,[0,1,2,3,4,5,...
1.0	4	1	2.0	[first, of, all, ...	[first, warning, ...	(20,[0,1,2,3,4,5,...	(20,[0,1,2,3,4,5,...
1.0	4	1	2.0	[imaginative, and...	[imaginative, dar...	(20,[0,3,4,5,6,7,...	(20,[0,3,4,5,6,7,...
1.0	5	0	1.0	[cannot, put, the...	[put, book, down....	(20,[1,4,7,8,9,10...	(20,[1,4,7,8,9,10...

Quantile Discretizer

Tokenizer

StopWords Remover

HashingTF

IDF

Models for Case 1&2: a snapshot

Use case 1: reviewText -> helpfulVotes
Use case 2: reviewText -> overall

Logistic Regression

Random Forest numTrees= 10

Random Forest numTrees= 100

```
# Set parameters for Logistic Regression
lgr2 = LogisticRegression(maxIter=10,
                          featuresCol = 'features',
                          labelCol='label')

# Fit the model to the data.
lgr2m = lgr2.fit(train_df)

# Given a dataset, predict each point's label, and show the results.
predictions2 = lgr2m.transform(test_df)
```

```
rf2 = RandomForestClassifier(labelCol="hvcat",
                             featuresCol="features",
                             numTrees=100,
                             maxDepth = 4,
                             maxBins = 32)
rf2m = rf2.fit(train_df)

rf2_pred = rf2m.transform(test_df)
```

Use Case 1 metrics, reviewText -> helpfulVotes

Using 0.01% of the data

Smaller Scale	accuracy	f1
Logistic	62.1%	60.5%
Random Forest 1	67.2%	67.2%
Random Forest 2	67.6%	67.4%

10% of the data ~1.2mn rows

Larger Scale	accuracy	f1
Logistic	63.7%	61.2%
Random Forest 1	65.8%	65.7%
Random Forest 2	65.4%	65.6%

80% train, 20% test

Use Case 2 metrics, reviewText -> overall

Using 0.01% of the data

Smaller Scale	accuracy	f1
Logistic	80.9%	72.3%
Random Forest 1	80.9%	72.3%
Random Forest 2	80.9%	72.3%

10% of the data ~1.2mn rows

Larger Scale	accuracy	f1
Logistic	83.0%	75.0%
Random Forest 1	82.8%	75.8%
Random Forest 2	82.3%	75.5%

80% train, 20% test

Insights & Next Steps

1. Explore more online storage options such as **AWS, GCP, MS Azure** and experience their in-house infrastructures.
2. Facilitate Big Data storage optimization and compression techniques.
3. Bringing in **more data** (volume wise) could **only** boost the ML performance to some degree. More focus should be put on **fine-tuning** the parameters.
4. Utilize other attributes to further compliment the ML so there are more attributes.
5. Insight: 1,000 reviews can give a good sense of what the whole dataset contains in terms of the text review data.
6. The distribution of the overall ratings, as well as helpfulVotes range did **not** change as we added more data.
7. Go watch Hasan Minhaj's take on Amazon