

EDA of Amazon Electronics Data with Pyspark and Spark SQL

In this Notebook, I will be performing Exploratory Data Analysis on Review and Product Meta data for Amazon Electronic products from 1996 to 2023. I will be writing queries using both Pyspark API and Spark SQL. In theory, the same types of queries can be written for either of them and they should have similar performance and run times. Queries will be written using both methods to verify that the same results can be obtained.

Setup

```
In [1]: from datasets import load_dataset
        from pyspark.sql import SparkSession
        from pyspark.sql import functions as F
        from pyspark.sql.types import (StructType, StructField, StringType, DoubleType, LongType, ArrayType, BooleanType)
        from pyspark import StorageLevel
        import pandas as pd
```

```
In [2]: reviews = load_dataset("McAuley-Lab/Amazon-Reviews-2023", "raw_review_Electronics", trust_remote_code=True)
        meta = load_dataset("McAuley-Lab/Amazon-Reviews-2023", "raw_meta_Electronics", split="full", trust_remote_code=True)
        reviews = reviews["full"]
```

Loading dataset shards: 0%| | 0/34 [00:00<?, ?it/s]

The data was obtained from <https://huggingface.co/datasets/McAuley-Lab/Amazon-Reviews-2023>

```
In [3]: print(type(reviews))
        print(type(meta))
```

```
<class 'datasets.arrow_dataset.Dataset'>
<class 'datasets.arrow_dataset.Dataset'>
```

```
In [4]: print(reviews.shape)
        print(meta.shape)
```

```
(43886944, 10)
(1610012, 16)
```

The reviews data contains over 43 million rows and 10 columns, while the meta data contains around 1.6 million rows and 16 columns. We will be using Pyspark to manipulate the data since Pandas cannot support data of this size.

```
import shutil
import os
```

```
# Save to Parquet (temporary folder)
reviews_path = "temp_reviews_parquet"
meta_path = "temp_meta_parquet"

reviews.to_parquet(reviews_path)
meta.to_parquet(meta_path)
```

Data is saved to Parquet for more efficient storage and processing.

```
# Clean up the Parquet folders
shutil.rmtree('temp_reviews_parquet')
shutil.rmtree('temp_meta_parquet')

print("Temporary Parquet files deleted.")
```

Create a SparkSession

```
In [5]: # Rename to Localhost
import os
os.environ["SPARK_LOCAL_HOSTNAME"] = "localhost"
```

```
In [6]: # Create the SparkSession
spark = SparkSession.builder \
    .appName('amazon_electronics_data') \
    .config('spark.driver.memory', '16g') \
    .getOrCreate()
spark
```

Out[6]: **SparkSession - in-memory**

SparkContext

[Spark UI](#)

Version	v3.5.5
Master	local[*]
AppName	amazon_electronics_data

```
In [7]: # Display Memory setting
spark.sparkContext.getConf().get("spark.driver.memory")
```

```
Out[7]: '16g'
```

A SparkSession needs to be created before further work with PySpark can be done. Since the data is around 10 gb in size, Spark's default 1g of memory is insufficient and queries would not run. 16g is set and we verify that this is the case.

Reviews Data

Parquet tables were previously created for the data for faster and more efficient manipulation of this large data.

```
# Direct Dataframe creation
reviews_spark_df = spark.createDataFrame(reviews)
```

The data is too large for direct dataframe creation.

```
# Set Schema
schema = StructType([
    StructField('rating', DoubleType(), True),
    StructField('title', StringType(), True),
    StructField('images', ArrayType(
        StructType([
            StructField("attachment_type", StringType(), True),
            StructField("large_image_url", StringType(), True),
            StructField("medium_image_url", StringType(), True),
            StructField("small_image_url", StringType(), True),
        ])
    ), True),
    StructField('asin', StringType(), True),
    StructField('parent_asin', StringType(), True),
    StructField('user_id', StringType(), True),
    StructField('timestamp', LongType(), True),
    StructField('helpful_vote', LongType(), True),
    StructField('verified_purchase', BooleanType(), True)
])

# Convert reviews dataset to pyarrow table
arrow_table = reviews.data

# Convert to list of Python dicts
records = arrow_table.to_pylist()
```

```
# Convert pyarrow table to Spark DataFrame
reviews_spark_df = spark.createDataFrame(records, schema=schema)
```

The data is too large for this as well. The data is saved to Parquet.

```
In [8]: # Read the Parquet file
reviews = spark.read.parquet("temp_reviews_parquet")

# Create the Spark Table and Temp View for querying with Pyspark API and Spark SQL
reviews.createOrReplaceTempView('reviews')
reviews = spark.table('reviews')
```

The data is read successfully. A table and temporary view were created to run queries.

Schema

```
In [9]: # Print Reviews Schema
reviews.printSchema()
```

```
root
|-- rating: double (nullable = true)
|-- title: string (nullable = true)
|-- text: string (nullable = true)
|-- images: array (nullable = true)
|   |-- element: struct (containsNull = true)
|   |   |-- attachment_type: string (nullable = true)
|   |   |-- large_image_url: string (nullable = true)
|   |   |-- medium_image_url: string (nullable = true)
|   |   |-- small_image_url: string (nullable = true)
|-- asin: string (nullable = true)
|-- parent_asin: string (nullable = true)
|-- user_id: string (nullable = true)
|-- timestamp: long (nullable = true)
|-- helpful_vote: long (nullable = true)
|-- verified_purchase: boolean (nullable = true)
```

This data is on the user reviews of the products. A review consists of a title and a body of text. The asin and parent_asin represent product identifiers. Some products have some minor differences (likely in options like size or color).

```
In [10]: # Create the Spark Table and Temp View for querying with Pyspark API and Spark SQL
reviews.createOrReplaceTempView('reviews')
reviews = spark.table('reviews')
```

Queries can be run using Spark SQL or Pyspark API. We'll be writing the queries for each method to verify that results are the same.

Show the First 10 Rows of the Data

```
In [11]: # With Pyspark API using .select()
reviews.select(*reviews.columns).limit(10).show(truncate=14)
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
----+
|rating|      title|      text|      images|      asin|parent_asin|      user_id|      timestamp|helpful_vote|verified_purc
hase|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
----+
|  3.0|Smells like...|First & mos...|[{IMAGE, ht...|B083NRGZMM| B083NRGZMM|AFKZENTNBQ7...|1658185117948|      0|
true|
|  1.0|Didn't work...|These didn'...|      []|B07N69T6TM| B07N69T6TM|AFKZENTNBQ7...|1592678549731|      0|
true|
|  5.0|    Excellent!|I love thes...|      []|B01G8J05F2| B01G8J05F2|AFKZENTNBQ7...|1523093017534|      0|
true|
|  5.0|Great lapto...|I was searc...|      []|B0010C5JKY| B0010C5JKY|AGGZ357A026...|1290278495000|     18|
true|
|  5.0|Best Headph...|I've bought...|      []|B013J7WUGC| B07CJYMRWM|AG2L7H23R5L...|1676601581238|      0|
true|
|  5.0|Great Fan! ...|Light weigh...|      []|B072DSHKCH| B07CML419K|AGCI7FAH4GL...|1637522881041|      0|
true|
|  5.0|solid sound...|Update 2-th...|      []|B07BHHB5RH| B07BHHB5RH|AGCI7FAH4GL...|1565130879386|      0|
true|
|  5.0|Love the he...|These are f...|      []|B07BND376H| B09S6Y5BRG|AGCI7FAH4GL...|1541356831659|      9|
true|
|  5.0|    Five Stars|pretty good...|      []|B002HWRZ2K| B01LW71IBJ|AGCI7FAH4GL...|1456772571000|      0|
true|
|  5.0|BUY THIS THANG|yes.. so go...|      []|B00WK47VEW| B017T99JPG|AGCI7FAH4GL...|1456772365000|      0|
true|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
----+
```

```
In [12]: # With Pyspark API omitting .select()
reviews.limit(10).show(truncate=14)
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|rating|      title|      text|      images|      asin|parent_asin|      user_id|      timestamp|helpful_vote|verified_purc|
|hase|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+----+
|  3.0|Smells like...|First & mos...|[{IMAGE, ht...|B083NRGZMM| B083NRGZMM|AFKZENTNBQ7...|1658185117948|      0|
true|
|  1.0|Didn't work...|These didn'...|      []|B07N69T6TM| B07N69T6TM|AFKZENTNBQ7...|1592678549731|      0|
true|
|  5.0|    Excellent!|I love thes...|      []|B01G8J05F2| B01G8J05F2|AFKZENTNBQ7...|1523093017534|      0|
true|
|  5.0|Great lapto...|I was searc...|      []|B0010C5JKY| B0010C5JKY|AGGZ357A026...|1290278495000|     18|
true|
|  5.0|Best Headph...|I've bought...|      []|B013J7WUGC| B07CJYMRWM|AG2L7H23R5L...|1676601581238|      0|
true|
|  5.0|Great Fan! ...|Light weigh...|      []|B072DSHKCH| B07CML419K|AGCI7FAH4GL...|1637522881041|      0|
true|
|  5.0|solid sound...|Update 2-th...|      []|B07BHHB5RH| B07BHHB5RH|AGCI7FAH4GL...|1565130879386|      0|
true|
|  5.0|Love the he...|These are f...|      []|B07BND376H| B09S6Y5BRG|AGCI7FAH4GL...|1541356831659|      9|
true|
|  5.0|    Five Stars|pretty good...|      []|B002HWRZ2K| B01LW71IBJ|AGCI7FAH4GL...|1456772571000|      0|
true|
|  5.0|BUY THIS THANG|yes.. so go...|      []|B00WK47VEW| B017T99JPG|AGCI7FAH4GL...|1456772365000|      0|
true|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+----+

```

```

In [13]: # With Spark SQL
spark.sql("""
SELECT *
FROM reviews
LIMIT 10
""").show(truncate=14)

```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|rating|      title|      text|      images|      asin|parent_asin|      user_id|      timestamp|helpful_vote|verified_purc
hase|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+----+
|  3.0|Smells like...|First & mos...|[{IMAGE, ht...|B083NRGZMM| B083NRGZMM|AFKZENTNBQ7...|1658185117948|      0|
true|
|  1.0|Didn't work...|These didn'...|      []|B07N69T6TM| B07N69T6TM|AFKZENTNBQ7...|1592678549731|      0|
true|
|  5.0|    Excellent!|I love thes...|      []|B01G8J05F2| B01G8J05F2|AFKZENTNBQ7...|1523093017534|      0|
true|
|  5.0|Great lapto...|I was searc...|      []|B0010C5JKY| B0010C5JKY|AGGZ357A026...|1290278495000|     18|
true|
|  5.0|Best Headph...|I've bought...|      []|B013J7WUGC| B07CJYMRWM|AG2L7H23R5L...|1676601581238|      0|
true|
|  5.0|Great Fan! ...|Light weigh...|      []|B072DSHKCH| B07CML419K|AGCI7FAH4GL...|1637522881041|      0|
true|
|  5.0|solid sound...|Update 2-th...|      []|B07BHHB5RH| B07BHHB5RH|AGCI7FAH4GL...|1565130879386|      0|
true|
|  5.0|Love the he...|These are f...|      []|B07BND376H| B09S6Y5BRG|AGCI7FAH4GL...|1541356831659|      9|
true|
|  5.0|    Five Stars|pretty good...|      []|B002HWRZ2K| B01LW71IBJ|AGCI7FAH4GL...|1456772571000|      0|
true|
|  5.0|BUY THIS THANG|yes.. so go...|      []|B00WK47VEW| B017T99JPG|AGCI7FAH4GL...|1456772365000|      0|
true|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+----+

```

The `.select()` part can be omitted when selecting all columns within PySpark API. Using Spark SQL we get the same results as when we use PySpark API.

Create a datetime column

We have a timestamp column using unix time using milliseconds. This is far less interpretable than something like datetime. Let's create a datetime column.

```

In [14]: # Add the datetime column then update the table
reviews = spark.table("reviews").withColumn("datetime", F.from_unixtime(F.col("timestamp") / 1000))

# Cache the data and trigger it
#reviews.persist(StorageLevel.DISK_ONLY)
#reviews.count()

```

```
# Update the view
reviews.createOrReplaceTempView("reviews")
```

Now we need to verify that the datetime column was added.

```
In [15]: # With Pyspark API
reviews.limit(10).show(truncate=14)
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
|rating|      title|      text|      images|      asin|parent_asin|      user_id|      timestamp|helpful_vote|verified_purc
hase|      datetime|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
|   3.0|Smells like...|First & mos...|[{IMAGE, ht...|B083NRGZMM| B083NRGZMM|AFKZENTNBQ7...|1658185117948|      0|
true|2022-07-18 ...|
|   1.0|Didn't work...|These didn'...|      []|B07N69T6TM| B07N69T6TM|AFKZENTNBQ7...|1592678549731|      0|
true|2020-06-20 ...|
|   5.0|   Excellent!|I love thes...|      []|B01G8J05F2| B01G8J05F2|AFKZENTNBQ7...|1523093017534|      0|
true|2018-04-07 ...|
|   5.0|Great lapto...|I was searc...|      []|B0010C5JKY| B0010C5JKY|AGGZ357A026...|1290278495000|     18|
true|2010-11-20 ...|
|   5.0|Best Headph...|I've bought...|      []|B013J7WUGC| B07CJYMRWM|AG2L7H23R5L...|1676601581238|      0|
true|2023-02-16 ...|
|   5.0|Great Fan! ...|Light weigh...|      []|B072DSHKCH| B07CML419K|AGCI7FAH4GL...|1637522881041|      0|
true|2021-11-21 ...|
|   5.0|solid sound...|Update 2-th...|      []|B07BHHB5RH| B07BHHB5RH|AGCI7FAH4GL...|1565130879386|      0|
true|2019-08-06 ...|
|   5.0|Love the he...|These are f...|      []|B07BND376H| B09S6Y5BRG|AGCI7FAH4GL...|1541356831659|      9|
true|2018-11-04 ...|
|   5.0|   Five Stars|pretty good...|      []|B002HWRZ2K| B01LW71IBJ|AGCI7FAH4GL...|1456772571000|      0|
true|2016-02-29 ...|
|   5.0|BUY THIS THANG|yes.. so go...|      []|B00WK47VEW| B017T99JPG|AGCI7FAH4GL...|1456772365000|      0|
true|2016-02-29 ...|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
```

```
In [16]: # With Spark SQL
spark.sql("""
SELECT *
FROM reviews
LIMIT 10
""").show(truncate=14)
```



```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|rating|      title|      text|      images|      asin|parent_asin|      user_id|      timestamp|helpful_vote|verified_purc
hase|      datetime|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  3.0|Smells like...|First & mos...|[{IMAGE, ht...|B083NRGZMM| B083NRGZMM|AFKZENTNBQ7...|1658185117948|      0|
true|2022-07-18 ...|
|  1.0|Didn't work...|These didn'...|      []|B07N69T6TM| B07N69T6TM|AFKZENTNBQ7...|1592678549731|      0|
true|2020-06-20 ...|
|  5.0|    Excellent!|I love thes...|      []|B01G8J05F2| B01G8J05F2|AFKZENTNBQ7...|1523093017534|      0|
true|2018-04-07 ...|
|  5.0|Great lapto...|I was searc...|      []|B0010C5JKY| B0010C5JKY|AGGZ357A026...|1290278495000|     18|
true|2010-11-20 ...|
|  5.0|Best Headph...|I've bought...|      []|B013J7WUGC| B07CJYMRWM|AG2L7H23R5L...|1676601581238|      0|
true|2023-02-16 ...|
|  5.0|Great Fan! ...|Light weigh...|      []|B072DSHKCH| B07CML419K|AGCI7FAH4GL...|1637522881041|      0|
true|2021-11-21 ...|
|  5.0|solid sound...|Update 2-th...|      []|B07BHHB5RH| B07BHHB5RH|AGCI7FAH4GL...|1565130879386|      0|
true|2019-08-06 ...|
|  5.0|Love the he...|These are f...|      []|B07BND376H| B09S6Y5BRG|AGCI7FAH4GL...|1541356831659|      9|
true|2018-11-04 ...|
|  5.0|    Five Stars|pretty good...|      []|B002HWRZ2K| B01LW71IBJ|AGCI7FAH4GL...|1456772571000|      0|
true|2016-02-29 ...|
|  5.0|BUY THIS THANG|yes.. so go...|      []|B00WK47VEW| B017T99JPG|AGCI7FAH4GL...|1456772365000|      0|
true|2016-02-29 ...|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+

```

Summary Statistics

```

In [17]: reviews\
          .select('rating', 'helpful_vote')\
          .summary('count', 'mean', 'stddev', 'min', '25%', '50%', '75%', 'max')\
          .show()

```

	summary	rating	helpful_vote
	count	43886944	43886944
	mean	4.099233772121385	1.0903363423983223
	stddev	1.4123304685968932	22.575534823738767
	min	0.0	-4
	25%	4.0	0
	50%	5.0	0
	75%	5.0	0
	max	5.0	46841

Here are the summary statistics for the numeric variables. The areas of interest are the min rating of 0.0 and the min helpful_vote of -4. The ratings of 0.0 should be incorrect as Amazon ratings go from 1 to 5. Amazon used to have not helpful votes, but later took out that feature for their product user reviews. That -4 may be from that time.

Null values

```
In [18]: # With Pyspark API
reviews\
.select([
    F.sum(F.when(F.col(c).isNull(), 1).otherwise(0)).alias(f"{c}_nulls")
    for c in reviews.columns
])\
.show()
```

	rating_nulls	title_nulls	text_nulls	images_nulls	asin_nulls	parent_asin_nulls	user_id_nulls	timestamp_nulls	helpful_vote_nulls	verified_purchase_nulls	datetime_nulls
	0	0	0	0	0	0	0	0	0	0	0

```
In [19]: # With Spark SQL

# Set the column names
cols = reviews.columns

# Generate SQL snippet for each column
```

```

null_checks = ",\n ".join(
    [f"SUM(CASE WHEN {c} IS NULL THEN 1 ELSE 0 END) AS `{c}_nulls`" for c in cols]
)

# Create the query
query = f"""
SELECT
    {null_checks}
FROM reviews
"""

# Run the query
spark.sql(query).show()

```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|rating_nulls|title_nulls|text_nulls|images_nulls|asin_nulls|parent_asin_nulls|user_id_nulls|timestamp_nulls|helpful_vote_nulls|verified_purchase_nulls|datetime_nulls|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|0|0|0|0|0|0|0|0|0|0|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

There are no null values in the reviews data.

Number of unique values for each column

```

In [20]: # With Pyspark API
reviews\
.select([
    F.countDistinct(c).alias(f"{c}_unique")
    for c in reviews.columns
])\
.show()

```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|rating_unique|title_unique|text_unique|images_unique|asin_unique|parent_asin_unique|user_id_unique|timestamp_unique|helpful_vote_u
nique|verified_purchase_unique|datetime_unique|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|          6|    19258533|   38250600|    1987253|    1946161|        1609860|        18286191|        42534478|
2102|                2|    39949939|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

In [21]: # With Spark SQL
         uniques = ",\n ".join(
             [f"COUNT(DISTINCT {c}) AS `{c}_unique`" for c in cols]
         )

         # Create the query
         query = f"""
         SELECT
             {uniques}
         FROM reviews
         """

         # Run the query
         spark.sql(query).show()

```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|rating_unique|title_unique|text_unique|images_unique|asin_unique|parent_asin_unique|user_id_unique|timestamp_unique|helpful_vote_u
nique|verified_purchase_unique|datetime_unique|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|          6|    19258533|   38250600|    1987253|    1946161|        1609860|        18286191|        42534478|
2102|                2|    39949939|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

A couple of interesting/notable things here:

We have 6 unique values for ratings which matches our previous finding of ratings being from 0.0 to 5.0. The only problematic value here is the 0.0.

There are 38 million unique values for 'text,' which represents the bodies for reviews. Given that the data has around 43 million observations, around 5 million reviews have the same text. If we were to check the reviews, we may find that they are short and generally applicable such as 'Good.'

The difference between 'asin' and 'parent_asin' is around 300,000. This value represents the number of products that appear to be the same but have different options like a shirt having different color options.

The number of unique verified purchases looks correct.

Verified purchases for each rating

How many reviews (split between verified purchases) do we have for each rating?

```
In [22]: # With PySpark API
reviews\
.groupBy("rating")\
.agg(
    F.sum(F.when(F.col("verified_purchase") == "true", 1).otherwise(0)).alias("Number of Verified Purchases"),
    F.sum(F.when(F.col("verified_purchase") == "false", 1).otherwise(0)).alias("Number of Non-Verified Purchases")
)\
.orderBy(F.col("rating").desc())\
.show()
```

rating	Number of Verified Purchases	Number of Non-Verified Purchases
5.0	26044266	1785482
4.0	5044312	513413
3.0	2625915	257150
2.0	2028015	229589
1.0	4804374	554426
0.0	2	0

```
In [23]: # With Spark SQL
spark.sql("""
SELECT
    rating,
    SUM(CASE WHEN verified_purchase = 'true' THEN 1 ELSE 0 END) AS `Number of Verified Purchases`,
    SUM(CASE WHEN verified_purchase = 'false' THEN 1 ELSE 0 END) AS `Number of Non-Verified Purchases`
FROM reviews
GROUP BY rating
ORDER BY rating DESC
""").show()
```

rating	Number of Verified Purchases	Number of Non-Verified Purchases
5.0	26044266	1785482
4.0	5044312	513413
3.0	2625915	257150
2.0	2028015	229589
1.0	4804374	554426
0.0	2	0

We previously saw that the min was 0.0 for ratings. Given that there are only 2 of them in the data, they are likely errors in data entry. We'll query and look at the 2 reviews in question. It is likely that the reviews had negative sentiment given the value is at the lower extreme for ratings.

Handling the reviews with ratings of 0

Let's query which rows have ratings of 0.

```
In [24]: # Verify that there are no reviews of 0.0 using PySpark API
reviews\
.filter(F.col('rating') == 0.0)\
.select('rating', 'title', 'text', 'parent_asin', 'user_id')\
.show(truncate=40)
```

rating	title	text	parent_asin	user_id
0.0	NOT UNIVERSAL	Didn't work for my device.	B09Z6Q2MLC	AHCTFZB5JDOEM6GKBWF4PDEXMCPA
0.0	One Star	I need a replacement the unit was dam...	B01F3ENAOU	AG5AKJ4EROK2F6BD4SCQVZ7WP7KQ

```
In [25]: # Verify that there are no reviews of 0.0 using Spark SQL
spark.sql("""
SELECT rating, title, text, parent_asin, user_id
FROM reviews
WHERE rating = 0.0
""").show(truncate=40)
```

rating	title	text	parent_asin	user_id
0.0	NOT UNIVERSAL	Didn't work for my device.	B09Z6Q2MLC	AHCTFZB5JDOEM6GKBWF4PDEXMCPA
0.0	One Star	I need a replacement the unit was dam...	B01F3ENAOU	AG5AKJ4EROK2F6BD4SCQVZ7WP7KQ

From the title and text, the reviews do indeed appear to have negative sentiment. Incidentally, the second one has a title of 'One Star.' Since the ratings were mistakenly entered as 0.0, let's correct them by setting them to 1.0.

```
In [26]: # Change those ratings to 1.0 and update the table
reviews = reviews.withColumn(
    "rating",
    F.when(F.col("rating") == 0.0, 1.0).otherwise(F.col("rating"))
)

# Update the View for SparkSQL
reviews.createOrReplaceTempView("reviews")
```

Now that the ratings of 0 have been changed, let's make sure there are no longer any ratings of 0 in the data.

```
In [27]: # Verify that there are no reviews of 0.0 using PySpark API
reviews\
    .filter(F.col('rating') == 0.0)\
    .select(F.count('rating'))\
    .show()
```

count(rating)
0

```
In [28]: # Verify that there are no reviews of 0.0 using Spark SQL
spark.sql("""
SELECT COUNT(rating)
FROM reviews
WHERE rating = 0.0
""").show()
```

```
+-----+
|count(rating)|
+-----+
|              0|
+-----+
```

We now have no reviews with ratings of 0.0 in the data. To check further, let's check the rows for those previous reviews, using the parent_asin and the user_id.

In [29]: *# Retrieve the rows that previously had ratings of 0.0 with PySpark API*

Save the query for user 1

```
user1 = reviews\
.filter(
    (F.col('parent_asin') == 'B09Z6Q2MLC')
    &
    (F.col('user_id') == 'AHCTFZB5JDOEM6GKBWF4PDEXMCPA')
)\
.select('rating', 'title', 'text', 'parent_asin', 'user_id')\
```

Save the query for user 2

```
user2 = reviews\
.filter(
    (F.col('parent_asin') == 'B01F3ENAOU')
    &
    (F.col('user_id') == 'AG5AKJ4EROK2F6BD4SCQVZ7WP7KQ')
)\
.select('rating', 'title', 'text', 'parent_asin', 'user_id')\
```

Perform a union and merge the two user tables, returning a table with both users

```
user1.union(user2).show(truncate=40)
```

```
+-----+-----+-----+-----+-----+
|rating|      title|              text|parent_asin|              user_id|
+-----+-----+-----+-----+-----+
|   1.0|NOT UNIVERSAL|    Didn't work for my device.| B09Z6Q2MLC|AHCTFZB5JDOEM6GKBWF4PDEXMCPA|
|   1.0|    One Star|I need a replacement the unit was dam...| B01F3ENAOU|AG5AKJ4EROK2F6BD4SCQVZ7WP7KQ|
+-----+-----+-----+-----+-----+
```

In [30]: *# Retrieve the rows that previously had ratings of 0.0 with Unions in Spark SQL*

```
spark.sql("""
SELECT rating, title, text, parent_asin, user_id
FROM reviews
WHERE parent_asin = 'B09Z6Q2MLC' AND user_id = 'AHCTFZB5JDOEM6GKBWF4PDEXMCPA'

UNION ALL
```



```
SELECT rating, title, text, parent_asin, user_id
FROM reviews
WHERE parent_asin = 'B01F3ENAOU' AND user_id = 'AG5AKJ4EROK2F6BD4SCQVZ7WP7KQ'
""").show(truncate = 40)
```

rating	title	text	parent_asin	user_id
1.0	NOT UNIVERSAL	Didn't work for my device.	B09Z6Q2MLC	AHCTFZB5JDOEM6GKBWF4PDEXMCPA
1.0	One Star	I need a replacement the unit was dam...	B01F3ENAOU	AG5AKJ4EROK2F6BD4SCQVZ7WP7KQ

The ratings have now been changed to 1 and the incorrect entries have been fixed.

Handling Negative Helpful Votes

Let's find all of the rows that have negative helpful votes.

```
In [31]: # With PySpark API
reviews\
.filter(F.col('helpful_vote') < 0)\
.orderBy('datetime')\
.show(truncate=14)
```

rating	title	text	images	asin	parent_asin	user_id	timestamp	helpful_vote	verified_purchase
5.0	Secure sour...	It's so hel...	[]	B078162HNC	B078162HNC	AFGXFHGZ7NV...	1547336082365	-1	true
9-01-12 ...									
5.0	Excellent R...	I have had ...	[]	B082X3H6P9	B082X3H6P9	AEVZGLYSS3C...	1589208570253	-4	true
0-05-11 ...									
5.0	Wow. Stunning.	Or, maybe i...	[]	B08CVQ5SD9	B09M8DYH1B	AGCSSPVU3FJ...	1619730920197	-2	true
1-04-29 ...									
5.0	Very Handy!	Just what I...	[]	B084GFMGN1	B0B2RK874T	AHFWJG5A5X2...	1625355873555	-1	true
1-07-03 ...									

```
In [32]: # With Spark SQL
spark.sql("""
SELECT *
```

```
FROM reviews
WHERE helpful_vote < 0
ORDER BY datetime
""").show(truncate=14)
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
|rating|      title|      text|images|      asin|parent_asin|      user_id|      timestamp|helpful_vote|verified_purchase|
datetime|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
|  5.0|Secure sour...|It's so hel...|  []|B078162HNC| B078162HNC|AFGXFHGZ7NV...|1547336082365|      -1|      true|201
9-01-12 ...|
|  5.0|Excellent R...|I have had ...|  []|B082X3H6P9| B082X3H6P9|AEVZGLYSS3C...|1589208570253|      -4|      true|202
0-05-11 ...|
|  5.0|Wow. Stunning.|Or, maybe i...|  []|B08CVQ5SD9| B09M8DYH1B|AGCSSPVU3FJ...|1619730920197|      -2|      true|202
1-04-29 ...|
|  5.0|  Very Handy!|Just what I...|  []|B084GFMGN1| B0B2RK874T|AHFWJG5A5X2...|1625355873555|      -1|      true|202
1-07-03 ...|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
```

Amazon used to have a 'not helpful vote' option, but that was later removed. There is no source that can be cited on the specific date, so we can't handle this in a targeted manner like leaving the values if the 'not helpful vote' option was there at that time. We will therefore change the negative values to 0.

Next, we will remove those rows from the reviews data.

```
In [33]: # Change those helpful votes to 0 and update the reviews table
reviews = reviews.withColumn(
    "helpful_vote",
    F.when(F.col("helpful_vote") < 0, 0).otherwise(F.col("helpful_vote"))
)

# Update the View for SparkSQL
reviews.createOrReplaceTempView("reviews")
```

Helpful Votes below 0 have been replaced and should now take values of 0. Next, we want to make sure that there are no longer any Helpful Votes below 0 in the data.

```
In [34]: # With PySpark API
reviews\
.filter(F.col('helpful_vote') < 0)\
```

```
.select(F.count('helpful_vote'))\
.show()
```

```
+-----+
|count(helpful_vote)|
+-----+
|                  0|
+-----+
```

```
In [35]: # With Spark SQL
spark.sql("""
SELECT COUNT(helpful_vote)
FROM reviews
WHERE helpful_vote < 0
""").show()
```

```
+-----+
|count(helpful_vote)|
+-----+
|                  0|
+-----+
```

From these queries, the handling went smoothly.

Number of verified and non-verified purchases

How many verified and non-verified purchases do we have in the data?

```
In [36]: # With Pyspark API
reviews.select(
    F.sum(F.when(F.col('verified_purchase') == True, 1).otherwise(0))
    .alias('Number of Verified Purchases'),
    F.sum(F.when(F.col('verified_purchase') == False, 1).otherwise(0))
    .alias('Number of Non-Verified Purchases')
).show()
```

```
+-----+-----+
|Number of Verified Purchases|Number of Non-Verified Purchases|
+-----+-----+
|          40546884          |          3340060          |
+-----+-----+
```

```
In [37]: # With Spark SQL
spark.sql("""
```

```
SELECT
SUM(CASE WHEN verified_purchase = 'true' THEN 1 ELSE 0 END) AS `Number of Verified Purchases`,
SUM(CASE WHEN verified_purchase = 'false' THEN 1 ELSE 0 END) AS `Number of Non-Verified Purchases`
FROM reviews
""").show()
```

```
+-----+-----+
|Number of Verified Purchases|Number of Non-Verified Purchases|
+-----+-----+
|                40546884|                3340060|
+-----+-----+
```

The number of non-verified purchases is around 10% that of the number of verified purchases.

Which verified reviews were the most helpful?

Among product reviews, what's the highest numbers of helpful votes, rating, and product identifier (parent asin) for those reviews with verified purchases?

```
In [38]: # With Pyspark API
reviews\
.filter(
    (F.col('helpful_vote') > 0)
    &
    (F.col('verified_purchase') == 'true')
)\
.select(
    'parent_asin',
    'rating',
    'title',
    'text',
    'verified_purchase',
    'helpful_vote'
)\
.orderBy('helpful_vote', ascending=False)\
.limit(15)\
.show(truncate=40)
```

parent_asin	rating	title	text	verified_purchase	helpful_vote
B010BWYDYA	5.0	Why and how the Kindle changes everyt...	This is less a "pros and cons" review...	true	4684
B075X8471B	5.0	What I once thought was silly is now ...	Earlier this month, my TV entertainme...	true	3294
B00CX5P8FC	5.0	This box is a GAME CHANGER for on dem...	I am not a casual user of on-demand c...	true	2633
B00154JDAI	1.0	BEWARE of the SIGNIFICANT DIFFERENCES...	I was DELIGHTED to upgrade my Kindle ...	true	2426
B07PHQ93TV	4.0	Not going to blow smoke...	I see so many people on here who are ...	true	1850
B01K8B8YA8	5.0	The Smartest of Them All!!! (Check ou...	[[VIDEOID:c05af0259c5236d5d0d0d5dea9b...	true	1814
B0070ZNZQ0	5.0	Amazing new Kindle is nearly perfect ...	So far, I love my new Paperwhite Kind...	true	1647
B008GG93YE	4.0	New Kindle Buyer? Thinking about Nook...	I got my first Kindle. This is a nice...	true	1516
B09KQC7Z4S	5.0	I give it five stars. My wife hates t...	I give it five stars. My wife hates t...	true	1507
B0BWD4WGJB	2.0	One major flaw.	I bought this to use as a baby monito...	true	1464
B010BWYDYA	5.0	This is a steal for \$50 as long as yo...	I pre-ordered this for my wife mostly...	true	1415
B07KTYJ769	5.0	Great smart plug to go with echo!	So I'm kind of new to the amazon devi...	true	1397
B0792K2BK6	4.0	Love the Dot and its new look, but th...	I've been a happy owner of a 2nd gene...	true	1396
B007Q1W586	5.0	My Thoughts (Typed Using The Kindle F...	Let me start out by saying that I am ...	true	1351
B0791TX5P5	5.0	The Remote quickly learns how to powe...	Although there are already 9,000 plus...	true	1345

```
In [39]: # With Spark SQL
spark.sql("""
SELECT parent_asin, rating, title, text, verified_purchase, helpful_vote
FROM reviews
WHERE helpful_vote > 0 AND verified_purchase = 'true'
ORDER BY helpful_vote DESC
```

```
LIMIT 15
""").show(truncate=40)
```

+-----+-----+-----+-----+-----+-----+					
-+					
parent_asin	rating	title	text	verified_purchase	helpful_vote
+-----+-----+-----+-----+-----+-----+					
-+					
B010BWYDYA	5.0	Why and how the Kindle changes everyt...	This is less a "pros and cons" review...	true	4684
1					
B075X8471B	5.0	What I once thought was silly is now ...	Earlier this month, my TV entertainme...	true	3294
8					
B00CX5P8FC	5.0	This box is a GAME CHANGER for on dem...	I am not a casual user of on-demand c...	true	2633
1					
B00154JDAI	1.0	BEWARE of the SIGNIFICANT DIFFERENCES...	I was DELIGHTED to upgrade my Kindle ...	true	2426
1					
B07PHQ93TV	4.0	Not going to blow smoke...	I see so many people on here who are ...	true	1850
1					
B01K8B8YA8	5.0	The Smartest of Them All!!! (Check ou...	[[VIDEOID:c05af0259c5236d5d0d0d5dea9b...	true	1814
7					
B0070ZNZQ0	5.0	Amazing new Kindle is nearly perfect ...	So far, I love my new Paperwhite Kind...	true	1647
8					
B008GG93YE	4.0	New Kindle Buyer? Thinking about Nook...	I got my first Kindle. This is a nice...	true	1516
9					
B09KQC7Z4S	5.0	I give it five stars. My wife hates t...	I give it five stars. My wife hates t...	true	1507
4					
B0BWD4WGJB	2.0	One major flaw.	I bought this to use as a baby monito...	true	1464
7					
B010BWYDYA	5.0	This is a steal for \$50 as long as yo...	I pre-ordered this for my wife mostly...	true	1415
3					
B07KTYJ769	5.0	Great smart plug to go with echo!	So I'm kind of new to the amazon devi...	true	1397
3					
B0792K2BK6	4.0	Love the Dot and its new look, but th...	I've been a happy owner of a 2nd gene...	true	1396
7					
B007Q1W586	5.0	My Thoughts (Typed Using The Kindle F...	Let me start out by saying that I am ...	true	1351
0					
B0791TX5P5	5.0	The Remote quickly learns how to powe...	Although there are already 9,000 plus...	true	1345
3					
+-----+-----+-----+-----+-----+-----+					
-+					

The most helpful product reviews with verified purchases tend to have high (4 or 5) ratings. Most of them are for different products. Those Helpful Vote counts above 18,000 are likely outliers.

Which non-verified reviews were the most helpful?

Among product reviews, what's the highest numbers of helpful votes, rating, and product identifier (parent asin) for those reviews without verified purchases?

```
In [40]: # With Pyspark API
reviews\
.filter(
    (F.col('helpful_vote') > 0)
    &
    (F.col('verified_purchase') == 'false')
)\
.select(
    'parent_asin',
    'rating',
    'title',
    'text',
    'verified_purchase',
    'helpful_vote'
)\
.orderBy('helpful_vote', ascending=False)\
.limit(15)\
.show(truncate=40)
```

	parent_asin	rating	title	text	verified_purchase	helpful_vote
7	B000I1X6PM	3.0	Rift in the time-space continuum	The minute I plugged this cable in, I...	false	1258
9	B00LWHUBP0	5.0	Detailed Review of the Kindle	As has been the case for years, Amazo...	false	1057
1	B08RLW7918	1.0	Spy alert!!!!	The camera video quality is definitel...	false	953
7	B000J36XR2	1.0	I have only a little time...	We live underground. We speak with ou...	false	895
6	B07ZPC9QD4	4.0	List of Features/Changes compared to ...	Like iphone 11 Pro, Macbook Pro, Airp...	false	862
8	B08M8Y6473	3.0	The Pros, Cons and Oks for Fitbit Ver...	Pros:_____ • It's very comf...	false	860
4	B009T3EYH0	4.0	A very good computer with a few drawb...	***Updates To My Review At The End***...	false	838
7	B07RJZPTLX	4.0	The Pros, Cons and Oks for Kindle Oas...	Pros:_____ • It's small, p...	false	797
0	B00L40308U	5.0	Financing Available!	I was able to purchase this amazing t...	false	766
2	B0BVY4JVNG	1.0	Only half the equation. The other hal...	I just received this today. <br ...	false	763
7	B08F6GPRH6	1.0	Not interested in monthly fee	Passed, not interested in paying \$120...	false	754
1	B0933BVK6T	4.0	The Pros, Cons and Oks for Apple's Ai...	Pros:_____ • No weight issu...	false	726
2	B00N2ZDXW2	3.0	Cool but three problems	First off, the Ring is very cool. I r...	false	708
8	B00N2ZDXW2	1.0	When put to the test, it failed at ev...	[[VIDEOID:57547f1d2d767853416bf63487f...	false	652
6	B0C2ZMJW53	5.0	a real opinion	I could sit here and write all about ...	false	638

```
In [41]: # With Spark SQL
spark.sql("""
SELECT parent_asin, rating, title, text, verified_purchase, helpful_vote
FROM reviews
WHERE helpful_vote > 0 AND verified_purchase = 'false'
ORDER BY helpful_vote DESC
```



```
LIMIT 15
""").show(truncate=40)
```

parent_asin	rating	title	text	verified_purchase	helpful_votes
B000I1X6PM	3.0	Rift in the time-space continuum	The minute I plugged this cable in, I...	false	1258
B00LWHUBPO	5.0	Detailed Review of the Kindle	As has been the case for years, Amazo...	false	1057
B08RLW7918	1.0	Spy alert!!!!	The camera video quality is definitel...	false	953
B000J36XR2	1.0	I have only a little time...	We live underground. We speak with ou...	false	895
B07ZPC9QD4	4.0	List of Features/Changes compared to ...	Like iphone 11 Pro, Macbook Pro, Airp...	false	862
B08M8Y6473	3.0	The Pros, Cons and Oks for Fitbit Ver...	Pros:_____ • It's very comf...	false	860
B009T3EYHO	4.0	A very good computer with a few drawb...	***Updates To My Review At The End***...	false	838
B07RJZPTLX	4.0	The Pros, Cons and Oks for Kindle Oas...	Pros:_____ • It's small, p...	false	797
B00L40308U	5.0	Financing Available!	I was able to purchase this amazing t...	false	766
B0BVY4JVNG	1.0	Only half the equation. The other hal...	I just received this today. <br ...	false	763
B08F6GPRH6	1.0	Not interested in monthly fee	Passed, not interested in paying \$120...	false	754
B0933BVK6T	4.0	The Pros, Cons and Oks for Apple's Ai...	Pros:_____ • No weight issu...	false	726
B00N2ZDXW2	3.0	Cool but three problems	First off, the Ring is very cool. I r...	false	708
B00N2ZDXW2	1.0	When put to the test, it failed at ev...	[[VIDEOID:57547f1d2d767853416bf63487f...	false	652
B0C2ZMJW53	5.0	a real opinion	I could sit here and write all about ...	false	638

Reviews that aren't verified purchases tend to have much fewer helpful votes than those that do. The top 15 Helpful Vote count for the verified purchases are close to double the top 15 Helpful Vote count for the non-verified purchases, if we exclude potential outliers.

Some of the reviews also have formatting as part of the text. There is also one review that has a video, and we can see how it's handled in the data: converted to a video id.

Verified purchases for each rating

How many verified and non-verified purchases do we have for each rating?

```
In [42]: # With Pyspark API
reviews\
  .groupBy('rating')\
  .agg(
    F.sum(F.when(F.col('verified_purchase') == True, 1).otherwise(0))
      .alias('Number of Verified Purchases'),
    F.sum(F.when(F.col('verified_purchase') == False, 1).otherwise(0))
      .alias('Number of Non-Verified Purchases')
  )\
  .orderBy('rating', ascending=False)\
  .show()
```

```
+-----+-----+-----+
|rating|Number of Verified Purchases|Number of Non-Verified Purchases|
+-----+-----+-----+
| 5.0|26044266|1785482|
| 4.0|5044312|513413|
| 3.0|2625915|257150|
| 2.0|2028015|229589|
| 1.0|4804376|554426|
+-----+-----+-----+
```

```
In [43]: # With Spark SQL
spark.sql("""
SELECT
  rating,
  SUM(CASE WHEN verified_purchase = 'true' THEN 1 ELSE 0 END) AS `Number of Verified Purchases`,
  SUM(CASE WHEN verified_purchase = 'false' THEN 1 ELSE 0 END) AS `Number of Non-Verified Purchases`
FROM reviews
GROUP BY rating
ORDER BY rating DESC
""").show()
```

rating	Number of Verified Purchases	Number of Non-Verified Purchases
5.0	26044266	1785482
4.0	5044312	513413
3.0	2625915	257150
2.0	2028015	229589
1.0	4804376	554426

Ratings between 2 to 4 have similar numbers if we were to account for the proportion of data. The number of non-verified purchases are around 10% the number of verified purchases for those ratings.

There tend to be more ratings of 5 for verified purchases and more ratings of 1 for non-verified purchases.

As expected, the ratings of 1 and 5 are where the difference lie. There may be more ratings from non-verified purchases due to review bombing from competitors.

How many of each rating was there in year 2023?

```
In [44]: # With PySpark API
reviews\
  .filter(F.year("datetime") == 2023) \
  .groupBy("rating") \
  .agg(
    F.count("*").alias("num_ratings_2023"),
    F.sum(F.when(F.col("verified_purchase") == "true", 1).otherwise(0)).alias("Number of Verified Purchases"),
    F.sum(F.when(F.col("verified_purchase") == "false", 1).otherwise(0)).alias("Number of Non-Verified Purchases")
  )\
  .orderBy("num_ratings_2023", ascending=False)\
  .show()
```

rating	num_ratings_2023	Number of Verified Purchases	Number of Non-Verified Purchases
5.0	1190958	1054998	135960
1.0	289692	268795	20897
4.0	198503	161945	36558
3.0	124327	109692	14635
2.0	105836	96773	9063

```
In [45]: # With Spark SQL
spark.sql("""
```

```

SELECT
    rating,
    COUNT(*) AS num_ratings_2023,
    SUM(CASE WHEN verified_purchase = 'true' THEN 1 ELSE 0 END) AS `Number of Verified Purchases`,
    SUM(CASE WHEN verified_purchase = 'false' THEN 1 ELSE 0 END) AS `Number of Non-Verified Purchases`
FROM reviews
WHERE YEAR(datetime) = 2023
GROUP BY rating
ORDER BY num_ratings_2023 DESC
""").show()

```

rating	num_ratings_2023	Number of Verified Purchases	Number of Non-Verified Purchases
5.0	1190958	1054998	135960
1.0	289692	268795	20897
4.0	198503	161945	36558
3.0	124327	109692	14635
2.0	105836	96773	9063

The number of 5.0 ratings is significantly higher than the numbers of other ratings. The second rating with the highest count is 1.0. It looks like reviewers on Amazon tend to gravitate to the extremes when rating Electronics products. The numbers could reflect how Amazon screens and removes fake reviews showing that their focus is on 1 star reviews instead of 5 star reviews. It could also mean that users are more likely to leave reviews on (Electronics) products that they are extremely pleased with.

Meta Data

Schema

```

In [46]: # Read the Parquet file
meta = spark.read.parquet("temp_meta_parquet")

# Print Meta Schema
meta.printSchema()

```

```

root
|-- main_category: string (nullable = true)
|-- title: string (nullable = true)
|-- average_rating: double (nullable = true)
|-- rating_number: long (nullable = true)
|-- features: array (nullable = true)
|   |-- element: string (containsNull = true)
|-- description: array (nullable = true)
|   |-- element: string (containsNull = true)
|-- price: string (nullable = true)
|-- images: struct (nullable = true)
|   |-- hi_res: array (nullable = true)
|   |   |-- element: string (containsNull = true)
|   |-- large: array (nullable = true)
|   |   |-- element: string (containsNull = true)
|   |-- thumb: array (nullable = true)
|   |   |-- element: string (containsNull = true)
|   |-- variant: array (nullable = true)
|   |   |-- element: string (containsNull = true)
|-- videos: struct (nullable = true)
|   |-- title: array (nullable = true)
|   |   |-- element: string (containsNull = true)
|   |-- url: array (nullable = true)
|   |   |-- element: string (containsNull = true)
|   |-- user_id: array (nullable = true)
|   |   |-- element: string (containsNull = true)
|-- store: string (nullable = true)
|-- categories: array (nullable = true)
|   |-- element: string (containsNull = true)
|-- details: string (nullable = true)
|-- parent_asin: string (nullable = true)
|-- bought_together: string (nullable = true)
|-- subtitle: string (nullable = true)
|-- author: string (nullable = true)

```

This data represents product metadata. It contains information on the product, name, features, and seller. Subtitle and author are rather strange for Electronics products, so we'll look into that later.

```

In [47]: # Create the Spark Table and Temp View for querying with Pyspark API and Spark SQL
meta.createOrReplaceTempView('meta')
meta = spark.table('meta')

```

Show the First 10 Rows of the Data

```
In [48]: # With Pyspark API
meta.limit(10).show(truncate=10)
```

main_category	title	average_rating	rating_number	features	description	price	images	videos	store	categories
Electronics	Apple iPhone 11 Pro Max (128GB) - Space Gray	4.5	1234	5G, 12MP camera, A13 Bionic	Apple iPhone 11 Pro Max (128GB) - Space Gray	999.00	[https://apple.com/iphone11-pro-max]	[https://apple.com/iphone11-pro-max]	Apple	Electronics>Smartphones
Electronics	Samsung Galaxy S20 FE (5G) - Starlight	4.3	987	5G, 64MP camera, 5000mAh	Samsung Galaxy S20 FE (5G) - Starlight	699.99	[https://samsung.com/galaxy-s20-fe]	[https://samsung.com/galaxy-s20-fe]	Samsung	Electronics>Smartphones
Electronics	Google Pixel 4 XL - Black	4.6	765	5G, 108MP camera, 5120x2880 display	Google Pixel 4 XL - Black	799.00	[https://google.com/pixel4-xl]	[https://google.com/pixel4-xl]	Google	Electronics>Smartphones
Electronics	OnePlus 8 Pro - Green	4.4	543	5G, 64MP camera, 65W fast charge	OnePlus 8 Pro - Green	699.99	[https://oneplus.com/oneplus8pro]	[https://oneplus.com/oneplus8pro]	OnePlus	Electronics>Smartphones
Electronics	Xiaomi Mi 10 Pro 5G - Black	4.2	432	5G, 108MP camera, 120Hz display	Xiaomi Mi 10 Pro 5G - Black	599.00	[https://xiaomi.com/mi10pro]	[https://xiaomi.com/mi10pro]	Xiaomi	Electronics>Smartphones
Electronics	Huawei P40 Pro - Black	4.7	321	5G, 50MP camera, 50W fast charge	Huawei P40 Pro - Black	899.00	[https://huawei.com/p40pro]	[https://huawei.com/p40pro]	Huawei	Electronics>Smartphones
Electronics	Motorola Moto Z3 Play - Blue	4.1	210	5G, 12.5MP camera, foldable	Motorola Moto Z3 Play - Blue	499.99	[https://motorola.com/motoz3play]	[https://motorola.com/motoz3play]	Motorola	Electronics>Smartphones
Electronics	Nokia 7.2 - Black	4.0	109	5G, 48MP camera, 4000mAh	Nokia 7.2 - Black	399.00	[https://nokia.com/nokia72]	[https://nokia.com/nokia72]	Nokia	Electronics>Smartphones
Electronics	BlackBerry KEY2 - Black	3.9	98	5G, 12MP camera, QWERTY keyboard	BlackBerry KEY2 - Black	349.99	[https://blackberry.com/key2]	[https://blackberry.com/key2]	BlackBerry	Electronics>Smartphones
Electronics	Alcatel 1S 5G - Blue	3.8	87	5G, 13MP camera, 5000mAh	Alcatel 1S 5G - Blue	299.00	[https://alcatel.com/alcatel1s5g]	[https://alcatel.com/alcatel1s5g]	Alcatel	Electronics>Smartphones

```
In [49]: # With Spark SQL
spark.sql("""
SELECT *
FROM meta
LIMIT 10
""").show(truncate = 10)
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|main_category|title|average_rating|rating_number|features|description|price|images|videos|store|categories|
details|parent_asin|bought_together|subtitle|author|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|All Ele...|FS-1051...|3.5|6|[]|[Telepo...|None|{[NULL]...|{[], []...|Fat Shark|[Electr...|{"D
ate ...|B00MCW7G9M|NULL|NULL|NULL|
|All Ele...|Ce-H22B...|5.0|1|[UPC: 6...|[HDMI I...|None|{[https...|{[], []...|SIIG|[Electr...|{"P
rodu...|B00YT6XQSE|NULL|NULL|NULL|
|Computers|Digi-Ta...|4.5|246|[WARNIN...|[]|19.99|{[https...|{[AL 2S...|Digi-Tatoo|[Electr...|{"B
rand...|B07SM135LS|NULL|NULL|NULL|
|AMAZON ...|NotoCit...|4.5|233|[NotoC...|[]|9.99|{[https...|{[], []...|NotoCity|[Electr...|
{"Date ...|B089CNGZCW|NULL|NULL|NULL|
|Cell Ph...|Motorol...|3.8|64|[New Dr...|[all Ge...|14.99|{[NULL,...|{[], []...|Verizon|[Electr...|{"P
rodu...|B004E2Z880|NULL|NULL|NULL|
|Sports ...|Raymari...|3.5|25|[Black ...|[Transf...|None|{[https...|{[], []...|Raymarine|[Electr...|{"I
tem ...|B00TX536EK|NULL|NULL|NULL|
|Cell Ph...|QGHX0 B...|4.4|707|[Person...|[Compat...|14.89|{[https...|{[], []...|QGHX0|[Electr...|{"P
acka...|B07BJ7ZZL7|NULL|NULL|NULL|
|Industr...|Protech...|3.8|6|[iPhone...|[This a...|None|{[NULL,...|{[], []...|ProTech|[Electr...|{"L
ight...|B005G9902U|NULL|NULL|NULL|
|Computers|MOSISO ...|5.0|2|[]|[]|None|{[https...|{[], []...|MOSISO|[Electr...|{"B
rand...|B01MCZP7RF|NULL|NULL|NULL|
|NULL|Fishfin...|4.4|86|[Made o...|[Afford...|10.99|{[NULL,...|{[Unbox...|Westlak...|[Electr...|{"I
tem ...|B00R6R82HS|NULL|NULL|NULL|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+

```

There's a lot of nulls for the last 3 columns. We should look into that later.

Summary Statistics

```

In [50]: # With PySpark API
meta\
.select('average_rating', 'rating_number', 'price')\
.summary('count', 'mean', 'stddev', 'min', '25%', '50%', '75%', 'max')\
.show(truncate=20)

```

summary	average_rating	rating_number	price
count	1610012	1610012	1610012
mean	4.0085408680183505	180.4833075778317	96.45767498849601
stddev	0.8331378499766278	2543.9800659979737	319.1737878683855
min	1.0	1	0.0
25%	3.6	3	11.99
50%	4.2	12	22.9
75%	4.6	49	59.99
max	5.0	1034896	-

There are 1,610,012 rows in the data.

The Average Rating column ranges in values from 1 to 5, which is reasonable. Unsurprisingly, Average Ratings of exactly 5 are extreme values that probably result from having very few reviews. Most of the Average Ratings are above 3 (around 4 according to the mean). The Rating Number has a - for the max, which is likely do it being a string as reported in the schema. The same goes for the price. Interestingly, most electronic products are below \$60 in price.

```
In [51]: # Cast the price in the meta table
meta = meta.withColumn('price', F.col('price').cast('double'))

# Update the View for SparkSql
meta.createOrReplaceTempView("meta")
```

```
In [52]: # With PySpark API
meta\
.select('average_rating', 'rating_number', 'price')\
.summary('count', 'mean', 'stddev', 'min', '25%', '50%', '75%', 'max')\
.show(truncate=20)
```

summary	average_rating	rating_number	price
count	1610012	1610012	526449
mean	4.0085408680183505	180.4833075778317	96.45767498849601
stddev	0.8331378499766278	2543.9800659979737	319.1737878683855
min	1.0	1	0.0
25%	3.6	3	11.99
50%	4.2	12	22.9
75%	4.6	49	59.99
max	5.0	1034896	44630.0

We now have 44630.0 instead of - for the max price.

Number of Nulls in the Data

```
In [53]: # With Pyspark API
meta.select([
    F.sum(F.when(F.col(c).isNull(), 1).otherwise(0)).alias(f"{c}")
    for c in meta.columns
]).show(truncate=10)
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|main_category|title|average_rating|rating_number|features|description| price|images|videos|store|categories|details|parent_asin|bought_together|subtitle| author|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|          106334|    0|          0|          0|    0|          0|1083563|    0|    0| 9520|          0|    0|          0|
1610012| 1609161|1609536|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

```
In [54]: # With Spark SQL

# Set the column names
cols = meta.columns

# Generate SQL snippet for each column
null_checks = ",\n ".join(
    [f"SUM(CASE WHEN {c} IS NULL THEN 1 ELSE 0 END) AS `{c}`" for c in cols]
)

# Create the query
query = f"""
SELECT
    {null_checks}
FROM meta
"""

# Run the query
spark.sql(query).show()
```

main_category	title	average_rating	rating_number	features	description	price	images	videos	store	categories	details	parent_asin	bought_together	subtitle	author
106334	0	0	0	0	0	1083563	0	0	9520	0	0	0	1610012	1609161	1609536

The number of nulls look good for most of the data. The store having nulls might be related to third party sellers. For main_category though, we will need further investigation. There may a general main_category that aggregates other electronics that don't fit into any specific category. It looks like bought_together, subtitle, and author are entirely or mostly null.

```
In [55]: # With PySpark API
# Store null counts
null_counts = meta.select(
    F.sum(F.when(F.col('bought_together').isNull(), 1).otherwise(0)).alias('bought_together_nulls'),
    F.sum(F.when(F.col('subtitle').isNull(), 1).otherwise(0)).alias('subtitle_nulls'),
    F.sum(F.when(F.col('author').isNull(), 1).otherwise(0)).alias('author_nulls')
)

# Store not-null counts
not_null_counts = meta.select(
    F.sum(F.when(F.col('bought_together').isNotNull(), 1).otherwise(0)).alias('bought_together_not_nulls'),
    F.sum(F.when(F.col('subtitle').isNotNull(), 1).otherwise(0)).alias('subtitle_not_nulls'),
    F.sum(F.when(F.col('author').isNotNull(), 1).otherwise(0)).alias('author_not_nulls')
)

# Join both results with crossJoin
null_counts.crossJoin(not_null_counts).show()
```

bought_together_nulls	subtitle_nulls	author_nulls	bought_together_not_nulls	subtitle_not_nulls	author_not_nulls
1610012	1609161	1609536	0	851	476

```
In [56]: # With SparkSQL
# Use Common Table Expressions to store the Null and Not-Null queries, then join them
spark.sql("""
WITH null_counts AS (
  SELECT
    SUM(CASE WHEN bought_together IS NULL THEN 1 ELSE 0 END) AS bought_together_nulls,
```

```

        SUM(CASE WHEN subtitle IS NULL THEN 1 ELSE 0 END) AS subtitle_nulls,
        SUM(CASE WHEN author IS NULL THEN 1 ELSE 0 END) AS author_nulls
    FROM meta
),
not_null_counts AS (
    SELECT
        SUM(CASE WHEN bought_together IS NOT NULL THEN 1 ELSE 0 END) AS bought_together_not_nulls,
        SUM(CASE WHEN subtitle IS NOT NULL THEN 1 ELSE 0 END) AS subtitle_not_nulls,
        SUM(CASE WHEN author IS NOT NULL THEN 1 ELSE 0 END) AS author_not_nulls
    FROM meta
)

SELECT *
FROM null_counts
JOIN not_null_counts
""").show()

```

bought_together_nulls	subtitle_nulls	author_nulls	bought_together_not_nulls	subtitle_not_nulls	author_not_nulls
1610012	1609161	1609536	0	851	476

It looks like the table is mostly populated with null values for these columns. Bought_together in particular is entirely null and should be removed entirely.

Investigating Authors

Let's write a query to see what data we have where we have observations for the authors.

```

In [57]: # Pyspark API
meta\
.filter(F.col('author').isNotNull())\
.select('main_category', 'title', 'subtitle', 'categories', 'author')\
.limit(5)\
.show(truncate=30)

```

```

+-----+
|main_category|          title|          subtitle|          categories|
author|
+-----+
|      Books|Practical Modern Basketball...|          3rd Edition|[Electronics, Accessories &...|{'avatar': 'https://m.m
edia...|
|      Books|Garmin GPSMAP 496 Qref Chec...|Spiral-bound - December 15,...|[Electronics, Car & Vehicle...|{'avatar': 'https://m.m
edia...|
|      Books| Sleepwalker: The Last Sandman|      Paperback - July 30, 2010|[Electronics, eBook Readers...|{'avatar': 'https://m.m
edia...|
|      Books|Disney Moana: Movie Theater...|      Hardcover - October 4, 2016|[Electronics, Video Project...|{'avatar': 'https://m.m
edia...|
|      Books|Estrella errante (Zodiaco) ...|      Paperback - October 1, 2017|[Electronics, Computers & A...|{'avatar': 'https://m.m
edia...|
+-----+
-----+

```

```

In [58]: # Spark SQL
spark.sql("""
SELECT main_category, title, subtitle, categories, author
FROM meta
WHERE author IS NOT NULL
LIMIT 5
""").show(truncate=30)

```

```

+-----+
|main_category|          title|          subtitle|          categories|
author|
+-----+
|      Books|Practical Modern Basketball...|          3rd Edition|[Electronics, Accessories &...|{'avatar': 'https://m.m
edia...|
|      Books|Garmin GPSMAP 496 Qref Chec...|Spiral-bound - December 15,...|[Electronics, Car & Vehicle...|{'avatar': 'https://m.m
edia...|
|      Books| Sleepwalker: The Last Sandman|      Paperback - July 30, 2010|[Electronics, eBook Readers...|{'avatar': 'https://m.m
edia...|
|      Books|Disney Moana: Movie Theater...|      Hardcover - October 4, 2016|[Electronics, Video Project...|{'avatar': 'https://m.m
edia...|
|      Books|Estrella errante (Zodiaco) ...|      Paperback - October 1, 2017|[Electronics, Computers & A...|{'avatar': 'https://m.m
edia...|
+-----+
-----+

```

It looks like there are books in the electronics product data. From the subtitles, some of these are physical books rather than eBooks from Kindle. Let's filter out the Books from the main_category and see what we get.

```
In [59]: # Pyspark API
meta\
.filter(
    F.col('author').isNotNull() &
    (F.col('main_category') != 'Books')
)\
.select('main_category', 'title', 'subtitle', 'categories', 'author')\
.limit(5)\
.show(truncate=30)
```

main_category	title	subtitle	categories	author
Buy a Kindle	Five Little Monkeys Storybo...	Kindle Edition	[Electronics, Computers & A...	{'avatar': 'https://m.media...
Buy a Kindle	HF Antenna Accessories (Ama...	Kindle Edition	[Electronics, Television & ...	{'avatar': 'https://m.media...
Buy a Kindle	An Orphan in the Snow: The ...	Kindle Edition	[Electronics, eBook Readers...	{'avatar': 'https://m.media...
Buy a Kindle	Sister Peters in Amsterdam	Kindle Edition	[Electronics, Accessories &...	{'avatar': 'https://m.media...
Buy a Kindle	Basher Five-Two	Kindle Edition	[Electronics, Car & Vehicle...	{'avatar': 'https://m.media...

```
In [60]: # Spark SQL
spark.sql("""
SELECT main_category, title, subtitle, categories, author
FROM meta
WHERE author IS NOT NULL AND main_category != 'Books'
LIMIT 5
""").show(truncate=30)
```

main_category	title	subtitle	categories	author
Buy a Kindle	Five Little Monkeys Storybo...	Kindle Edition	[Electronics, Computers & A...	{'avatar': 'https://m.media...
Buy a Kindle	HF Antenna Accessories (Ama...	Kindle Edition	[Electronics, Television & ...	{'avatar': 'https://m.media...
Buy a Kindle	An Orphan in the Snow: The ...	Kindle Edition	[Electronics, eBook Readers...	{'avatar': 'https://m.media...
Buy a Kindle	Sister Peters in Amsterdam	Kindle Edition	[Electronics, Accessories &...	{'avatar': 'https://m.media...
Buy a Kindle	Basher Five-Two	Kindle Edition	[Electronics, Car & Vehicle...	{'avatar': 'https://m.media...

Now we really have Kindle books. Let's further remove these with filters and see if there is any other data.

```
In [61]: # Pyspark API
meta\
```

```

.filter(
  F.col('author').isNull() &
  (F.col('main_category') != 'Books') &
  (F.col('main_category') != 'Buy a Kindle')

)\
.select('main_category', 'title', 'subtitle', 'categories', 'author')\
.limit(5)\
.show(truncate=30)

```

```

+-----+-----+-----+-----+-----+
|main_category|title|subtitle|categories|author|
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+

```

```

In [62]: # Spark SQL
spark.sql("""
SELECT main_category, title, subtitle, categories, author
FROM meta
WHERE author IS NOT NULL AND main_category != 'Books' AND main_category != 'Buy a Kindle'
""").show(truncate=30)

```

```

+-----+-----+-----+-----+-----+
|main_category|title|subtitle|categories|author|
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+

```

So, the authors column has values only for products in the Books or Buy a Kindle main_categories. Since books aren't really electronics let's remove these products from the data.

Removing Books and Kindle eBooks

```

In [63]: # Removing the targeted rows with Pyspark API
meta = meta\
  .filter(
    ~F.col('main_category').isin('Books', 'Buy a Kindle')
  )

# Replace the view for SparkSql with the updated table
meta.createOrReplaceTempView("meta")

```

```

In [64]: # Double check that the rows were removed as intended using PySpark API
meta\
  .filter(

```

```
F.col('main_category').isin('Books', 'Buy a Kindle')
)\
.select('main_category', 'title', 'subtitle', 'categories', 'author')\
.limit(5)\
.show(truncate=30)
```

```
+-----+-----+-----+-----+-----+
|main_category|title|subtitle|categories|author|
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
```

In [65]: *# Double check that the rows were removed as intended using Spark SQL*

```
spark.sql("""
SELECT main_category, title, subtitle, categories, author
FROM meta
WHERE author IS NOT NULL AND main_category != 'Books' AND main_category != 'Buy a Kindle'
""").show(truncate=30)
```

```
+-----+-----+-----+-----+-----+
|main_category|title|subtitle|categories|author|
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
```

The rows were removed as intended. Next, let's check the number of nulls for the last 3 columns of the data table.

In [66]: *# With PySpark API*

Store null counts

```
null_counts = meta.select(
    F.sum(F.when(F.col('bought_together').isNull(), 1).otherwise(0)).alias('bought_together_nulls'),
    F.sum(F.when(F.col('subtitle').isNull(), 1).otherwise(0)).alias('subtitle_nulls'),
    F.sum(F.when(F.col('author').isNull(), 1).otherwise(0)).alias('author_nulls')
)
```

Store not-null counts

```
not_null_counts = meta.select(
    F.sum(F.when(F.col('bought_together').isNotNull(), 1).otherwise(0)).alias('bought_together_not_nulls'),
    F.sum(F.when(F.col('subtitle').isNotNull(), 1).otherwise(0)).alias('subtitle_not_nulls'),
    F.sum(F.when(F.col('author').isNotNull(), 1).otherwise(0)).alias('author_not_nulls')
)
```

Join both results with crossJoin

```
null_counts.crossJoin(not_null_counts).show()
```

bought_together_nulls	subtitle_nulls	author_nulls	bought_together_not_nulls	subtitle_not_nulls	author_not_nulls
1502617	1502617	1502617	0	0	0

```
In [67]: # With SparkSQL
# Use Common Table Expressions to store the Null and Not-Null queries, then join them
spark.sql("""
WITH null_counts AS (
    SELECT
        SUM(CASE WHEN bought_together IS NULL THEN 1 ELSE 0 END) AS bought_together_nulls,
        SUM(CASE WHEN subtitle IS NULL THEN 1 ELSE 0 END) AS subtitle_nulls,
        SUM(CASE WHEN author IS NULL THEN 1 ELSE 0 END) AS author_nulls
    FROM meta
),
not_null_counts AS (
    SELECT
        SUM(CASE WHEN bought_together IS NOT NULL THEN 1 ELSE 0 END) AS bought_together_not_nulls,
        SUM(CASE WHEN subtitle IS NOT NULL THEN 1 ELSE 0 END) AS subtitle_not_nulls,
        SUM(CASE WHEN author IS NOT NULL THEN 1 ELSE 0 END) AS author_not_nulls
    FROM meta
)

SELECT *
FROM null_counts
JOIN not_null_counts
""").show()
```

bought_together_nulls	subtitle_nulls	author_nulls	bought_together_not_nulls	subtitle_not_nulls	author_not_nulls
1502617	1502617	1502617	0	0	0

Now that the books were removed, the subtitle and author columns are entirely null.

Remove columns

Let's remove the bought_together, subtitle, and author columns since they are completely null. Let's remove the images and videos columns too.

```
In [68]: # Remove the bought_together, subtitle, image, and videos columns
meta = meta.drop('bought_together', 'subtitle', 'author', 'images', 'videos')
meta.createOrReplaceTempView("meta")
```


Now let's examine the resulting data table.

```
In [69]: # Print first 10 rows with PySpark API
meta.limit(10).show(truncate = 10)
```

main_category	title	average_rating	rating_number	features	description	price	store	categories	details	parent_asin
All Ele...	FS-1051...	3.5	6	[]	[Telepo...	NULL	Fat Shark	[Electr...	{"Date ...	B00MCW7G9M
All Ele...	Ce-H22B...	5.0	1	[UPC: 6...	[HDMI I...	NULL	SIIG	[Electr...	{"Produ...	B00YT6XQSE
Computers	Digi-Ta...	4.5	246	[WARNIN...	[]	19.99	Digi-Tatoo	[Electr...	{"Brand...	B07SM135LS
AMAZON ...	NotoCit...	4.5	233	[NotoC...	[]	9.99	NotoCity	[Electr...	{"Date ...	B089CNGZCW
Cell Ph...	Motorol...	3.8	64	[New Dr...	[all Ge...	14.99	Verizon	[Electr...	{"Produ...	B004E2Z880
Sports ...	Raymari...	3.5	25	[Black ...	[Transf...	NULL	Raymarine	[Electr...	{"Item ...	B00TX536EK
Cell Ph...	QGHXO B...	4.4	707	[Person...	[Compat...	14.89	QGHXO	[Electr...	{"Packa...	B07BJ7ZZL7
Industr...	Protech...	3.8	6	[iPhone...	[This a...	NULL	ProTech	[Electr...	{"Light...	B005G9902U
Computers	MOSISO ...	5.0	2	[]	[]	NULL	MOSISO	[Electr...	{"Brand...	B01MCZP7RF
Computers	Network...	3.3	14	[Connec...	[Produc...	NULL	Avanquest	[Electr...	{"Brand...	B000LV7P8I

```
In [70]: # Print first 10 rows with Spark SQL
spark.sql("""
SELECT *
FROM meta
LIMIT 10
""").show(truncate = 10)
```

main_category	title	average_rating	rating_number	features	description	price	store	categories	details	parent_asin
All Ele...	FS-1051...	3.5	6	[]	[Telepo...	NULL	Fat Shark	[Electr...	{"Date ...	B00MCW7G9M
All Ele...	Ce-H22B...	5.0	1	[UPC: 6...	[HDMI I...	NULL	SIIG	[Electr...	{"Produ...	B00YT6XQSE
Computers	Digi-Ta...	4.5	246	[WARNIN...	[]	19.99	Digi-Tatoo	[Electr...	{"Brand...	B07SM135LS
AMAZON ...	NotoCit...	4.5	233	[NotoC...	[]	9.99	NotoCity	[Electr...	{"Date ...	B089CNGZCW
Cell Ph...	Motorol...	3.8	64	[New Dr...	[all Ge...	14.99	Verizon	[Electr...	{"Produ...	B004E2Z880
Sports ...	Raymari...	3.5	25	[Black ...	[Transf...	NULL	Raymarine	[Electr...	{"Item ...	B00TX536EK
Cell Ph...	QGHXO B...	4.4	707	[Person...	[Compat...	14.89	QGHXO	[Electr...	{"Packa...	B07BJ7ZZL7
Industr...	Protech...	3.8	6	[iPhone...	[This a...	NULL	ProTech	[Electr...	{"Light...	B005G9902U
Computers	MOSISO ...	5.0	2	[]	[]	NULL	MOSISO	[Electr...	{"Brand...	B01MCZP7RF
Computers	Network...	3.3	14	[Connec...	[Produc...	NULL	Avanquest	[Electr...	{"Brand...	B000LV7P8I

The columns were removed as intended.

Highest Rating Numbers

```
In [71]: # With PySpark API
meta\
.select(
    F.col('parent_asin'),
    F.col('title'),
    F.col('rating_number'),
    F.col('main_category'),
)\
.orderBy('rating_number', ascending=False)\
.limit(10)\
.show(truncate=100)
```

parent_asin	title	rating_number
B07H65KP63	Echo Dot (3rd Gen, 2018 release) - Smart speaker with Alexa - Charcoal	1034896
B07GZFM1ZM	Fire TV Stick 4K streaming device with Alexa Voice Remote (includes TV controls) Dolby Vision	819630
B07PXGQC1Q	Apple AirPods (2nd Generation) Wireless Earbuds with Lightning Charging Case Included. Over 24 Ho...	585624
B08F1P3BCC	Echo Dot (4th Gen) Smart speaker with Alexa Twilight Blue	584328
B07KTYJ769	Amazon Smart Plug, for home automation, Works with Alexa - A Certified for Humans Device	542825
B0BGNG1294	Amazon Basics HDMI Cable, 18Gbps High-Speed, 4K@60Hz, 2160p, Ethernet Ready, 10 Foot, Black	507202
B0791TX5P5	Fire TV Stick streaming device with Alexa built in, includes Alexa Voice Remote, HD, latest release	474536
B08CLNX58K	SanDisk 32GB 2-Pack Ultra MicroSDHC UHS-I Memory Card (2x32GB) - SDSQUAR-032G-GN6MT	402800
B08WJSHSLC	Fire TV Stick (3rd Gen) with Alexa Voice Remote (includes TV controls) + Star Wars The Mandaloria...	379565
B07HZLHPKP	Echo Show 5 (1st Gen, 2019 release) -- Smart display with Alexa - stay connected with video calli...	356248

```
In [72]: # With SparkSQL
spark.sql("""
```

```
SELECT
parent_asin, title, rating_number, main_category
FROM meta
ORDER BY rating_number DESC
LIMIT 10
""").show(truncate=100)
```

parent_asin	title	rating_number
B07H65KP63	Echo Dot (3rd Gen, 2018 release) - Smart speaker with Alexa - Charcoal	1034896
B07GZFM1ZM	Fire TV Stick 4K streaming device with Alexa Voice Remote (includes TV controls) Dolby Vision	819630
B07PXGQC1Q	Apple AirPods (2nd Generation) Wireless Earbuds with Lightning Charging Case Included. Over 24 Ho...	585624
B08F1P3BCC	Echo Dot (4th Gen) Smart speaker with Alexa Twilight Blue	584328
B07KTYJ769	Amazon Smart Plug, for home automation, Works with Alexa - A Certified for Humans Device	542825
B0BGNG1294	Amazon Basics HDMI Cable, 18Gbps High-Speed, 4K@60Hz, 2160p, Ethernet Ready, 10 Foot, Black	507202
B0791TX5P5	Fire TV Stick streaming device with Alexa built in, includes Alexa Voice Remote, HD, latest release	474536
B08CLNX58K	SanDisk 32GB 2-Pack Ultra MicroSDHC UHS-I Memory Card (2x32GB) - SDSQUAR-032G-GN6MT	402800
B08WJSHSLC	Fire TV Stick (3rd Gen) with Alexa Voice Remote (includes TV controls) + Star Wars The Mandaloria...	379565
B07HZLHPKP	Echo Show 5 (1st Gen, 2019 release) -- Smart display with Alexa - stay connected with video calli...	356248

These are the products with the largest Rating Numbers. It does make sense that these products have a lot of ratings. The Echo Dot 3rd Gen has been discontinued, but an old page for it (<https://a.co/d/eRzCkEj>) shows around a million ratings and the numbers look legitimate.

Highest Prices

```
In [73]: # With PySpark API
meta\
.select(
    F.col('parent_asin'),
```

```

F.col('title'),
F.col('price'),
F.col('main_category'),
)\
.orderBy('price', ascending=False)\
.limit(10)\
.show(truncate=100)

```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
|parent_asin|                                     title|    price|
main_category|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
| B01N1P1685|                               Panasonic PT DS20K2U - SXGA+ DLP Projector - 20000 lumens| 30999.0|
All Electronics|
| B08VWPYY6W|                               Intelligent Security Patrol Service Robot Autonomous Navigation Robot (Black)| 26888.0|Industri
al & Scientific|
| B0B3S78JFL|LG 97-Inch Class OLED evo Gallery Edition G2 Series Alexa Built-in 4K Smart TV, 120Hz Refresh Rat...|24996.99|
All Electronics|
| B01NBNX2NP|                               Sony VPLVZ1000ES Ultra-Short Throw 4K HDR Home Theatre Projector| 24995.0|
All Electronics|
| B074Z53X99|                               Sony VPLVW885ES 4K HDR Laser Home Theater Video Projector|23390.12|
All Electronics|
| B07TS6J1DN|                               Wacom Intuos Small Refurbished Graphics Drawing Tablet|21111.35|
Computers|
| B01GGNKUS6|                               Panasonic 4K UHD 2160P LED-Backlit LCD Flat Panel Display 97.5" Black (TH-98LQ70LU)| 19974.0|
Computers|
| B07GM3J6SH|                               Sony 4K HDR Laser Home Theater Video Projector (VPLVW995ES)|19458.93|
All Electronics|
| B07HY8HTKR|                               JVC DLA-NX9 4K Home Theater Projector with 8K/e-Shift|17999.99|
All Electronics|
| B00WF9GZV2|                               HP 5412R 92GT PoE+/4SFP+ (No PSU) v3 z12 Switch (JL001A)|17937.16|
Computers|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+

```

```

In [74]: # With SparkSQL
spark.sql("""
SELECT
parent_asin, title, price, main_category
FROM meta
ORDER BY price DESC
LIMIT 10
""").show(truncate=100)

```

		title	price
parent_asin			
main_category			
B01N1P1685	Panasonic PT DS20K2U - SXGA+ DLP Projector - 20000 lumens	30999.0	
All Electronics			
B08VWPYY6W	Intelligent Security Patrol Service Robot Autonomous Navigation Robot (Black)	26888.0	Industri
al & Scientific			
B0B3S78JFL	LG 97-Inch Class OLED evo Gallery Edition G2 Series Alexa Built-in 4K Smart TV, 120Hz Refresh Rat...	24996.99	
All Electronics			
B01NBNX2NP	Sony VPLVZ1000ES Ultra-Short Throw 4K HDR Home Theatre Projector	24995.0	
All Electronics			
B074Z53X99	Sony VPLVW885ES 4K HDR Laser Home Theater Video Projector	23390.12	
All Electronics			
B07TS6J1DN	Wacom Intuos Small Refurbished Graphics Drawing Tablet	21111.35	
Computers			
B01GGNKUS6	Panasonic 4K UHD 2160P LED-Backlit LCD Flat Panel Display 97.5" Black (TH-98LQ70LU)	19974.0	
Computers			
B07GM3J6SH	Sony 4K HDR Laser Home Theater Video Projector (VPLVW995ES)	19458.93	
All Electronics			
B07HY8HTKR	JVC DLA-NX9 4K Home Theater Projector with 8K/e-Shift	17999.99	
All Electronics			
B00WF9GZV2	HP 5412R 92GT PoE+/4SFP+ (No PSU) v3 z12 Switch (JL001A)	17937.16	
Computers			

The priciest electronics products are mainly projectors. Other items include massive tvs, a high-end tablet for artists, and a network switch from HP.

Lowest Product Prices

```
In [75]: # With PySpark API
meta\
.select(
    F.col('parent_asin'),
    F.col('title'),
    F.col('price'),
    F.col('main_category'),
)\
.orderBy('price', ascending=True)\
.limit(10)\
.show(truncate=100)
```


parent_asin	main_category	title	price
B081XZLDZT	Cute VSCO Stickers for Water Bottles 50 Pcs Vinyl Waterproof Cool Stickers for Laptop, Luggage, P...	NULL	
B07H5DCZM2	Monster Truck Backdrop Car Racing Background Grave Digger 7x5ft Photo Booth Studio Props ZYVV0677	NULL	C
B09PYZQC2T	Lenovo IdeaPad 1 14IGL05 81VU00D3US 14" Notebook - Full HD - 1920 x 1080 - Intel Celeron N4020 Du...	NULL	
B07C1B94TT	HDMI Female to DV-D Male Rotating ADPTR	NULL	Al
B08DWT6XM5	Elejolie Vertical Holder for Laptop, Surface Vertical Laptop Stand 3 Slot with Adjustable Size, Alu...	NULL	Of
B0034VA88Q	Brand 250GB Hard Disk Drive/HDD for HP Pavilion DV4-1220 DV4-1222 dv4-1020 dv4-1028 dv4-1117 dv4-...	NULL	
B01LVVJPD3	SEH00 3 in 1 Lens, Fill Light, 140 Degree Wide Angle, 15X Macro Lens Clip-on Cell Phone Camera Le...	NULL	C
B00AYB9PUQ	Samsung Galaxy Tab 2 Garnet Red Edition Bundle with Case (7-Inch, Wi-Fi)	NULL	
B001R4Y8RY	Panduit NK5E88MEIY Category-5E 8-Wire Lead-Frame Style Jack Module, Electric Ivory, 4-Pair	NULL	Industrial
B01FML1QYA	Set of 8 Heavy-Duty Cord Carry Strap Handle & Hanger - Organize Cords, Hoses, Ropes (Set of 8)	NULL	Tools & Hom

We've got a lot of null prices. There doesn't seem to be a good way to handle them other than to ignore their prices or remove them entirely. Imputing prices would have to be done for each product individually, which someone with a data entry role could be assigned to do. Let's query again, but this time remove the nulls from consideration.

```
In [77]: # With PySpark API
meta\
.filter(
    F.col('price').isNotNull()
)\
.select(
    F.col('parent_asin'),
    F.col('title'),
    F.col('price'),
    F.col('main_category'),
)\
.orderBy('price', ascending=True)\
```


main_category	cat_count
Computers	418868
All Electronics	376435
Camera & Photo	223690
Cell Phones & Accessories	138237
Home Audio & Theater	106516
Industrial & Scientific	50555
Car Electronics	27332
Tools & Home Improvement	24042
Amazon Home	21299
Office Products	21250
AMAZON FASHION	18391
Sports & Outdoors	14581
Automotive	13956
GPS & Navigation	9339
Amazon Devices	8484
Portable Audio & Accessories	8050
Musical Instruments	6668
Toys & Games	3944
Health & Personal Care	3073
All Beauty	1848

```
In [80]: # With Spark SQL
spark.sql("""
SELECT DISTINCT
    main_category,
    COUNT(parent_asin) AS cat_count
FROM meta
GROUP BY main_category
ORDER BY cat_count DESC
LIMIT 20
""").show(truncate = False)
```

main_category	cat_count
Computers	418868
All Electronics	376435
Camera & Photo	223690
Cell Phones & Accessories	138237
Home Audio & Theater	106516
Industrial & Scientific	50555
Car Electronics	27332
Tools & Home Improvement	24042
Amazon Home	21299
Office Products	21250
AMAZON FASHION	18391
Sports & Outdoors	14581
Automotive	13956
GPS & Navigation	9339
Amazon Devices	8484
Portable Audio & Accessories	8050
Musical Instruments	6668
Toys & Games	3944
Health & Personal Care	3073
All Beauty	1848

The categories look mostly reasonable. It's hard to imagine what electronics products there are for Amazon Fashion though. Computers, cameras, cell phones, and home audio & theatre are the most common products listed on Amazon.

Which products had the highest average rating and how many ratings did they have?

```
In [81]: # With PySpark API
meta\
.select(
    'title',
    'parent_asin',
    'average_rating',
    'rating_number',
    'main_category'
)\
.orderBy('average_rating', ascending=False)\
.limit(10)\
.show(truncate = 75)
```

ain_category	title parent_asin average_rating rating_number	m
Dell 2023 Inspiron 16 7620 2-in-1 16" Touchscreen FHD+ Business Laptop, ...	B08SJJG6ZX	5.0 2
Computers		
Alpine SWR-1542D 15" 2000W Competition Car Audio Subwoofer+Vented Sub En...	B019RMRQ0I	5.0 1 Cell Phones &
Accessories		
HQRP AC Power Cord Compatible with Horizon Fitness SMT6.1P (TM113B) T10K...	B0BX77HKV4	5.0 2 All
Electronics		
HP 733688-001 HP New Keyboard ZBOOK	B014VTVKJA	5.0 2
Computers		
for iPod Touch Gen 5 - Gym Hair Don't Care - Gymnastic - Funny - Hipster	B014W9EBG8	5.0 1 Cell Phones &
Accessories		
Acer Aspire Laptop 5349-2592 Intel Celeron B800 1.5GHz 2GB 250GB DVDR/RW...	B007GY9DH4	5.0 2
Computers		
Bower ANP53072 Conversion Adapter Tube for Nikon Coolpix L830 Camera (72...	B00LUY7LDY	5.0 2 Ca
mera & Photo		
Kathy Mall Hello Kitty Cute Pink Creative Wired Usb Optical Mouse for De...	B00JX401I6	5.0 2
Computers		
Wharfedale 17" Tall Speaker Stands for Linton Speakers, Pair, Mahogany	B07YF7S3CV	5.0 4 All
Electronics		
1TB SSHD Solid State Hybrid Drive for Dell XPS-15 XPS-15 (L501x) XPS-15 ...	B0768QTGVL	5.0 5
Computers		

```
In [82]: # With Spark SQL
spark.sql("""
SELECT
    title,
    parent_asin,
    average_rating,
    rating_number,
    main_category
FROM meta
ORDER BY average_rating DESC
LIMIT 10
""").show(truncate = 75)
```

n_category	title parent_asin average_rating rating_number	mai
Yadda-Yadda Design Co. Palm Tree Moon - Laptop Vinyl Decal Sticker - Cop...	B00NSAI0P6	5.0 4 All E
electronics		
Diginex Bluetooth Earbuds Wireless Magnetic Headset Sport Earphones for ...	B00CXZIRC8	5.0 1
Computers		
HQR AC Power Cord Compatible with Horizon Fitness SMT6.1P (TM113B) T10K...	B0BX77HKV4	5.0 2 All E
electronics		
Senders 10pcs 2GB USB 2.0 Flash Drive Memory Stick Metal Key Shaped Memo...	B018U2KU7W	5.0 4
Computers		
CWK Laptop Charger AC Adpater Power Supply Cord Plug for Gateway ID49C, ...	B00S9GEXT6	5.0 2
Computers		
Fir, Weird 4XL Trees Cars qvc Clothes Small Pillow, Ganz red Stamp Irish...	B00Y06TEBM	5.0 1 All E
electronics		
Bower ANP53072 Conversion Adapter Tube for Nikon Coolpix L830 Camera (72...	B00LUY7LDY	5.0 2 Came
ra & Photo		
Cooler Fan for Delta ASB0912H 92mm x 25mm Quiet High Airflow Fan 18" Lon...	B086C6FZH2	5.0 1 Industrial &
Scientific		
HP 14 2020 Newest Business Laptop Computer I 14" FHD IPS I 10th Gen Inte...	B08B41M4WS	5.0 1
Computers		
PK Power Car DC Adapter for S Cobra 360 Degree 9 Band SCobra Radar Detec...	B06ZZMDQ4B	5.0 1 All E
electronics		

Unsurprisingly, the products with the highest average ratings (at the maximum value of 5) have very few ratings. Something similar will probably appear for those with the lowest ratings.

What are the most popular products? (measured by number of reviews)

```
In [83]: # With PySpark API
meta\
.select(
    'title',
    'parent_asin',
    'average_rating',
    'rating_number',
    'main_category'
)\
.orderBy('rating_number', ascending=False)\
```

```
.limit(10)\
.show(truncate=75)
```

category	title	parent_asin	average_rating	rating_number	main_c
Devices	Echo Dot (3rd Gen, 2018 release) - Smart speaker with Alexa - Charcoal	B07H65KP63	4.7	1034896	Amazon
Devices	Fire TV Stick 4K streaming device with Alexa Voice Remote (includes TV c...	B07GZFM1ZM	4.7	819630	Amazon
products	Apple AirPods (2nd Generation) Wireless Earbuds with Lightning Charging ...	B07PXGQC1Q	4.7	585624	Apple P
Devices	Echo Dot (4th Gen) Smart speaker with Alexa Twilight Blue	B08F1P3BCC	4.7	584328	Amazon
Devices	Amazon Smart Plug, for home automation, Works with Alexa - A Certified f...	B07KTYJ769	4.7	542825	Amazon
Theater	Amazon Basics HDMI Cable, 18Gbps High-Speed, 4K@60Hz, 2160p, Ethernet Re...	B0BGNG1294	4.7	507202	Home Audio &
Devices	Fire TV Stick streaming device with Alexa built in, includes Alexa Voice...	B0791TX5P5	4.7	474536	Amazon
mputers	SanDisk 32GB 2-Pack Ultra MicroSDHC UHS-I Memory Card (2x32GB) - SDSQUAR...	B08CLNX58K	4.7	402800	Co
Devices	Fire TV Stick (3rd Gen) with Alexa Voice Remote (includes TV controls) +...	B08WJSHSLC	4.7	379565	Amazon
Devices	Echo Show 5 (1st Gen, 2019 release) -- Smart display with Alexa - stay c...	B07HZLHPKP	4.6	356248	Amazon

```
In [84]: # With Spark SQL
spark.sql("""
SELECT
    title,
    parent_asin,
    average_rating,
    rating_number,
    main_category
FROM meta
ORDER BY rating_number DESC
LIMIT 10
""").show(truncate = 75)
```

category	title parent_asin average_rating rating_number	main_c
Devices	Echo Dot (3rd Gen, 2018 release) - Smart speaker with Alexa - Charcoal B07H65KP63	4.7 1034896 Amazon
Devices	Fire TV Stick 4K streaming device with Alexa Voice Remote (includes TV c... B07GZFM1ZM	4.7 819630 Amazon
roducts	Apple AirPods (2nd Generation) Wireless Earbuds with Lightning Charging ... B07PXGQC1Q	4.7 585624 Apple P
Devices	Echo Dot (4th Gen) Smart speaker with Alexa Twilight Blue B08F1P3BCC	4.7 584328 Amazon
Devices	Amazon Smart Plug, for home automation, Works with Alexa - A Certified f... B07KTYJ769	4.7 542825 Amazon
Theater	Amazon Basics HDMI Cable, 18Gbps High-Speed, 4K@60Hz, 2160p, Ethernet Re... B0BGNG1294	4.7 507202 Home Audio &
Devices	Fire TV Stick streaming device with Alexa built in, includes Alexa Voice... B0791TX5P5	4.7 474536 Amazon
mputers	SanDisk 32GB 2-Pack Ultra MicroSDHC UHS-I Memory Card (2x32GB) - SDSQUAR... B08CLNX58K	4.7 402800 Co
Devices	Fire TV Stick (3rd Gen) with Alexa Voice Remote (includes TV controls) +... B08WJSHSLC	4.7 379565 Amazon
Devices	Echo Show 5 (1st Gen, 2019 release) -- Smart display with Alexa - stay c... B07HZLHPKP	4.6 356248 Amazon

This time, the products are ordered by the rating_number.

The products with the most ratings also tend to have very high average ratings.

Amazon devices, such as Echo Dots and Fire Sticks, are extremely popular.

Most Popular Products Excluding Amazon Devices

```
In [85]: # With PySpark API
meta\
.filter(
    F.col('main_category') != 'Amazon Devices'
)\
.select(
    'title',
    'parent_asin',
```

```

        'average_rating',
        'rating_number',
        'main_category'
    )\
    .orderBy('rating_number', ascending=False)\
    .limit(10)\
    .show(truncate = 75)

```

```

+-----+-----+-----+-----+-----+
|                                     title|parent_asin|average_rating|rating_number|      main_c
category|
+-----+-----+-----+-----+-----+
|Apple AirPods (2nd Generation) Wireless Earbuds with Lightning Charging ...| B07PXGQC1Q|      4.7|      585624|      Apple P
roducts|
|Amazon Basics HDMI Cable, 18Gbps High-Speed, 4K@60Hz, 2160p, Ethernet Re...| B0BGNG1294|      4.7|      507202|Home Audio &
Theater|
|SanDisk 32GB 2-Pack Ultra MicroSDHC UHS-I Memory Card (2x32GB) - SDSQUAR...| B08CLNX58K|      4.7|      402800|      Co
mputers|
|TOZO T10 Bluetooth 5.3 Wireless Earbuds with Wireless Charging Case IPX8...| B08XPWDSWW|      4.3|      349254|Home Audio &
Theater|
|SanDisk 256GB Extreme microSDXC UHS-I Memory Card with Adapter - Up to 1...| B09V1FT19S|      4.8|      337225|      Co
mputers|
|                                     Apple AirPods Pro| B07ZPC9QD4|      4.8|      300544|      Apple P
roducts|
|               SanDisk Cruzer Blade 16GB USB 2.0 Flash Drive| B0BSFT117F|      4.7|      260446|      All Elec
tronics|
|TOZO T6 True Wireless Earbuds Bluetooth 5.3 Headphones Touch Control wit...| B08XNCHTCY|      4.4|      249207|      All Elec
tronics|
|SanDisk 32GB Ultra MicroSDHC UHS-I Memory Card with Adapter - 98MB/s, C1...| B00EKPHOY6|      4.7|      234767|      Co
mputers|
|SanDisk 32GB Ultra MicroSDHC UHS-I Memory Card with Adapter - 98MB/s, C1...| B073JWXGNT|      4.7|      233718|      Co
mputers|
+-----+-----+-----+-----+-----+
-----+

```

```

In [86]: # With Spark SQL
spark.sql("""
SELECT
    title,
    parent_asin,
    average_rating,
    rating_number,
    main_category
FROM meta
WHERE main_category != 'Amazon Devices'

```



```
ORDER BY rating_number DESC
LIMIT 10
""").show(truncate = 75)
```

category	title	parent_asin	average_rating	rating_number	main_c
Apple AirPods (2nd Generation) Wireless Earbuds with Lightning Charging ...		B07PXGQC1Q	4.7	585624	Apple P
Amazon Basics HDMI Cable, 18Gbps High-Speed, 4K@60Hz, 2160p, Ethernet Re...		B0BGNG1294	4.7	507202	Home Audio &
SanDisk 32GB 2-Pack Ultra MicroSDHC UHS-I Memory Card (2x32GB) - SDSQUAR...		B08CLNX58K	4.7	402800	Co
TOZO T10 Bluetooth 5.3 Wireless Earbuds with Wireless Charging Case IPX8...		B08XPWDSWW	4.3	349254	Home Audio &
SanDisk 256GB Extreme microSDXC UHS-I Memory Card with Adapter - Up to 1...		B09V1FT19S	4.8	337225	Co
Apple AirPods Pro		B07ZPC9QD4	4.8	300544	Apple P
SanDisk Cruzer Blade 16GB USB 2.0 Flash Drive		B0BSFT117F	4.7	260446	All Elec
TOZO T6 True Wireless Earbuds Bluetooth 5.3 Headphones Touch Control wit...		B08XNCHTCY	4.4	249207	All Elec
SanDisk 32GB Ultra MicroSDHC UHS-I Memory Card with Adapter - 98MB/s, C1...		B00EKPHOY6	4.7	234767	Co
SanDisk 32GB Ultra MicroSDHC UHS-I Memory Card with Adapter - 98MB/s, C1...		B073JWXGNT	4.7	233718	Co

There is still an Amazon Product in this table, so let's exclude also the items with 'Amazon' in their titles. Also, there are some products with the same title, although they may have different parent_asin identifiers. Let's restrict to unique values as well.

```
In [87]: meta.filter(
    (F.col("main_category") != "Amazon Devices") &
    (~F.col("title").like("Amazon%"))
)\
.groupBy("title", "main_category")\
.agg(
    F.max("parent_asin").alias("parent_asin"),
    F.max("average_rating").alias("average_rating"),
    F.max("rating_number").alias("rating_number")
)\
```

```
.dropDuplicates(["title", "main_category", "parent_asin", "average_rating", "rating_number"])\n.orderBy(F.col("rating_number").desc())\n.show(truncate=85)
```

ing rating_number	title	main_category parent_asin average_rat
Apple AirPods (2nd Generation) Wireless Earbuds with Lightning Charging Case Inclu...		Apple Products B07PXGQC1Q
4.7 585624		
SanDisk 32GB 2-Pack Ultra MicroSDHC UHS-I Memory Card (2x32GB) - SDSQUAR-032G-GN6MT		Computers B08CLNX58K
4.7 402800		
TOZO T10 Bluetooth 5.3 Wireless Earbuds with Wireless Charging Case IPX8 Waterproo...	Home Audio & Theater	B08XPWDSWW
4.3 349254		
SanDisk 256GB Extreme microSDXC UHS-I Memory Card with Adapter - Up to 160MB/s, C1...		Computers B09V1FT19S
4.8 337225		
Apple AirPods Pro		Apple Products B07ZPC9QD4
4.8 300544		
SanDisk Cruzer Blade 16GB USB 2.0 Flash Drive		All Electronics B0BSFT117F
4.7 260446		
TOZO T6 True Wireless Earbuds Bluetooth 5.3 Headphones Touch Control with Wireless...		All Electronics B08XNCHTCY
4.4 249207		
SanDisk 32GB Ultra MicroSDHC UHS-I Memory Card with Adapter - 98MB/s, C10, U1, Ful...		Computers B073JWXGNT
4.7 234767		
SanDisk 256GB Ultra microSDXC UHS-I Memory Card with Adapter - 120MB/s, C10, U1, F...		Computers B0BRZG5TK4
4.7 230266		
Apple Lightning to 3.5 mm Headphone Jack Adapter		Apple Products B0787QLV3H
4.7 223181		
SanDisk 128GB Ultra microSDXC UHS-I Memory Card with Adapter - 100MB/s, C10, U1, F...		Computers B073JYC4XM
4.7 218045		
Apple EarPods Headphones with Lightning Connector. Microphone with Built-in Remote...		Apple Products B01M0GB8CC
4.6 201075		
WD 2TB Elements Portable HDD, External Hard Drive, USB 3.0 for PC & Mac, Plug and ...		Computers B09ZQ4GHM8
4.7 200186		
Kingston 1.92TB A400 SATA 3 2.5" Internal SSD SA400S37/1920G - HDD Replacement for...		Computers B08CKZ36N7
4.8 193561		
OontZ Angle 3 Bluetooth Speaker, Portable Wireless Bluetooth 5.0 Speaker, 10 Watts...		All Electronics B0BW4PFM58
4.5 190291		
SABRENT 10 Port 60W USB 3.0 Hub with Individual Power Switches and LEDs Includes 6...		Computers B0BN74ZJDK
4.6 184507		
Apple 20W USB-C Power Adapter - iPhone Charger with Fast Charging Capability, Type...		Apple Products B08L5M9BTJ
4.8 183519		
Toshiba Canvio Basics 1TB Portable External Hard Drive 2.5 Inch USB 3.0 - Black - ...		Computers B07SPXSPZ7
4.6 179993		
Roku Express HD Roku Streaming Device with Simple Remote (no TV controls), Free ...		All Electronics B094PMZB6V
4.7 177618		
Fujifilm INSTAX Mini Instant Film Twin Pack (White) - International Version		Camera & Photo B07KPLTMYB
4.7 168774		

only showing top 20 rows

```
In [88]: # With Spark SQL
spark.sql("""
SELECT DISTINCT
    title,
    max(parent_asin) AS parent_asin,
    max(average_rating) AS average_rating,
    max(rating_number) AS rating_number,
    main_category
FROM meta
WHERE main_category != 'Amazon Devices' AND title NOT LIKE 'Amazon%'
GROUP BY title, main_category
ORDER BY rating_number DESC
""").show(truncate = 85)
```

main_category	title parent_asin average_rating rating_number		
Apple AirPods (2nd Generation) Wireless Earbuds with Lightning Charging Case Inclu... Apple Products	B07PXGQC1Q	4.7	585624
SanDisk 32GB 2-Pack Ultra MicroSDHC UHS-I Memory Card (2x32GB) - SDSQUAR-032G-GN6MT Computers	B08CLNX58K	4.7	402800
TOZO T10 Bluetooth 5.3 Wireless Earbuds with Wireless Charging Case IPX8 Waterproo... e Audio & Theater	B08XPWDSWW	4.3	349254 Hom
SanDisk 256GB Extreme microSDXC UHS-I Memory Card with Adapter - Up to 160MB/s, C1... Computers	B09V1FT19S	4.8	337225
 Apple AirPods Pro Apple Products	B07ZPC9QD4	4.8	300544
 SanDisk Cruzer Blade 16GB USB 2.0 Flash Drive All Electronics	B0BSFT117F	4.7	260446
TOZO T6 True Wireless Earbuds Bluetooth 5.3 Headphones Touch Control with Wireless... All Electronics	B08XNCHTCY	4.4	249207
SanDisk 32GB Ultra MicroSDHC UHS-I Memory Card with Adapter - 98MB/s, C10, U1, Ful... Computers	B073JWXGNT	4.7	234767
SanDisk 256GB Ultra microSDXC UHS-I Memory Card with Adapter - 120MB/s, C10, U1, F... Computers	B0BRZG5TK4	4.7	230266
 Apple Lightning to 3.5 mm Headphone Jack Adapter Apple Products	B0787QLV3H	4.7	223181
SanDisk 128GB Ultra microSDXC UHS-I Memory Card with Adapter - 100MB/s, C10, U1, F... Computers	B073JYC4XM	4.7	218045
Apple EarPods Headphones with Lightning Connector. Microphone with Built-in Remote... Apple Products	B01M0GB8CC	4.6	201075
WD 2TB Elements Portable HDD, External Hard Drive, USB 3.0 for PC & Mac, Plug and ... Computers	B09ZQ4GHM8	4.7	200186
Kingston 1.92TB A400 SATA 3 2.5" Internal SSD SA400S37/1920G - HDD Replacement for... Computers	B08CKZ36N7	4.8	193561
OontZ Angle 3 Bluetooth Speaker, Portable Wireless Bluetooth 5.0 Speaker, 10 Watts... All Electronics	B0BW4PFM58	4.5	190291
SABRENT 10 Port 60W USB 3.0 Hub with Individual Power Switches and LEDs Includes 6... Computers	B0BN74ZJDK	4.6	184507
Apple 20W USB-C Power Adapter - iPhone Charger with Fast Charging Capability, Type... Apple Products	B08L5M9BTJ	4.8	183519
Toshiba Canvio Basics 1TB Portable External Hard Drive 2.5 Inch USB 3.0 - Black - ... Computers	B07SPXSPZ7	4.6	179993
Roku Express HD Roku Streaming Device with Simple Remote (no TV controls), Free ... All Electronics	B094PMZB6V	4.7	177618
 Fujifilm INSTAX Mini Instant Film Twin Pack (White) - International Version Camera & Photo	B07KPLTMYB	4.7	168774

only showing top 20 rows

The most popular products seem to be Computer products and Apple products.

The most popular Apple products are accessories, and among the Apple products, audio devices like the AirPods are the most popular.

Of the Computer Products, memory cards and hard drives are the most popular.

Join Tables

```
In [89]: # Alias the DataFrames
r = reviews.alias('r')
m = meta.alias('m')

# Perform the Left join
joined_table = r.join(
    m,
    on='parent_asin',
    how='left'
)

# Select and rename columns from the aliased DataFrames
joined_table = joined_table.select(
    *[F.col(f'r.{col}') for col in reviews.columns],
    F.col('m.main_category'),
    F.col('m.title').alias('product_name'),
    F.col('m.price'),
    F.col('m.average_rating').alias('avg_product_rating'),
    F.col('m.categories'),
    F.col('m.description'),
    F.col('m.details')
)

# Show result
joined_table.limit(10).show(truncate=10)
```

rating	title	text	images	asin	parent_asin	user_id	timestamp	helpful_vote	verified_purchase	datetime	main_category
product_name	price	avg_product_rating	categories	description		details					
5.0	Nice!	I was p...	[]	B00068E91M	B00068E91M	AH0ZS6X...	1224633...	8	true	2008-10...	Car
Ele...	Cobra C...	41.44	4.3	[Electr...	[Cobra'...	{ "Item ...					
4.0	Has flows	I have ...	[]	B000VCLCHW	B000VCLCHW	AEH6UNO...	1402677...	0	false	2014-06...	Com
puters	Sony VA...	NULL	3.7	[Electr...	[Produc...	{ "Stand...					
5.0	Fair pr...	Fit per...	[]	B0015A8Q7S	B0015A8Q7S	AFYE72D...	1619277...	0	true	2021-04...	Car
Ele...	Scosche...	13.99	4.5	[Car & ...	[The Sc...	{ "Item ...					
5.0	Rangers...	This sk...	[]	B003S9XLW4	B003S9XLW4	AEY6ZB7...	1380835...	0	true	2013-10...	Cell
Ph...	Skinit ...	NULL	5.0	[Electr...	[Skinit...	{ "Produ...					
5.0	This RAM	This ra...	[]	B007CZDS64	B007CZDS64	AGRZIDR...	1387374...	3	true	2013-12...	Com
puters	G.SKILL...	NULL	4.0	[Electr...	[Model,...	{ "RAM":...					
2.0	NOT for...	I says ...	[]	B009NI8JRG	B009NI8JRG	AGCL4RC...	1505818...	0	true	2017-09...	Com
puters	Incase ...	9.99	2.8	[Electr...	[]	{ "Brand...					
2.0	Don't w...	Neither...	[]	B00BWF5U0M	B00BWF5U0M	AFX452C...	1602123...	0	true	2020-10...	Com
puters	SABRENT...	8.99	4.6	[Electr...	[]	{ "Wirel...					
5.0	Fantast...	Running...	[]	B00BWF5U0M	B00BWF5U0M	AF7N32B...	1376428...	6	true	2013-08...	Com
puters	SABRENT...	8.99	4.6	[Electr...	[]	{ "Wirel...					
4.0	Four Stars	Was as ...	[]	B00C1VEDJU	B00C1VEDJU	AF7UNSE...	1435787...	0	true	2015-07...	Com
puters	Case-on...	NULL	4.2	[Electr...	[Case-o...	{ "Produ...					
4.0	Two fac...	Two fac...	[]	B00DZT96C6	B00DZT96C6	AHDRQJG...	1462447...	0	true	2016-05...	Com
puters	Acer G2...	NULL	4.4	[Electr...	[Acer G...	{ "Stand...					

In [90]: *# Left Join and create a view with Spark SQL*

```
spark.sql("""
CREATE OR REPLACE TEMP VIEW joined_view AS
SELECT
    r.*,
    m.main_category,
    m.title AS product_name,
    m.price,
    m.average_rating AS avg_product_rating,
    m.categories,
    m.description,
    m.details
FROM reviews r
LEFT JOIN meta m
USING(parent_asin)
""")
```

Out[90]: DataFrame[]

In [91]: *# Query the first 10 rows of the joined_view table*

```
spark.sql("""
SELECT *
FROM joined_view
LIMIT 10
""").show(truncate=10)
```

parent_asin	rating	title	text	images	asin	user_id	timestamp	helpful_vote	verified_purchase	datetime	main_category
B00068E91M	5.0	Nice!	I was p...	[]	B00068E91M	AHOZS6X...	1224633...	8	true	2008-10...	Car
B000VCLCHW	4.0	Has flows	I have ...	[]	B000VCLCHW	AEH6UNO...	1402677...	0	false	2014-06...	Com
B0015A8Q7S	5.0	Fair pr...	Fit per...	[]	B0015A8Q7S	AFYE72D...	1619277...	0	true	2021-04...	Car
B003S9XLW4	5.0	Rangers...	This sk...	[]	B003S9XLW4	AEY6ZB7...	1380835...	0	true	2013-10...	Cell
B007CZDS64	5.0	This RAM	This ra...	[]	B007CZDS64	AGRZIDR...	1387374...	3	true	2013-12...	Com
B009NI8JRG	2.0	NOT for...	I says ...	[]	B009NI8JRG	AGCL4RC...	1505818...	0	true	2017-09...	Com
B00BWF5U0M	2.0	Don't w...	Neither...	[]	B00BWF5U0M	AFX452C...	1602123...	0	true	2020-10...	Com
B00BWF5U0M	5.0	Fantast...	Running...	[]	B00BWF5U0M	AF7N32B...	1376428...	6	true	2013-08...	Com
B00C1VEDJU	4.0	Four Stars	Was as ...	[]	B00C1VEDJU	AF7UNSE...	1435787...	0	true	2015-07...	Com
B00DZT96C6	4.0	Two fac...	Two fac...	[]	B00DZT96C6	AHDRQJG...	1462447...	0	true	2016-05...	Com

In 2023, what were the most popular products (counting only verified purchases)?

In [92]: *# With PySpark API*

```
joined_table\
.filter(
    (F.year('datetime') == 2023) &
    (F.col('verified_purchase') == 'true') &
```



```
(F.col('rating') == 5.0)
)\
.groupBy(
  'parent_asin',
  'product_name',
  F.year('datetime').alias('year')
)\
.agg(F.count('*').alias('5_star_verified_purchase_reviews'))\
.orderBy(F.col('5_star_verified_purchase_reviews').desc())\
.limit(10)\
.show(truncate=85)
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
|parent_asin|                                     product_name|year|5_star_verified_purchase_re
views|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
| B0BZ6XH3LC|ZIUTY Wireless Earbuds, Bluetooth 5.3 Headphones 50H Playtime with LED Digital Dis...|2023|
2976|
| B0BNZ2785V|TAGRY Bluetooth Headphones True Wireless Earbuds 60H Playback LED Power Display Ea...|2023|
2953|
| B0C1ZHJRGB|Wireless Earbuds Bluetooth Headphones 48hrs Play Back Sport Earphones with LED Dis...|2023|
2298|
| B0B4JQY8JY|Sunany USB Flash Drive 256 GB for Phone and Pad, High Speed External Thumb Drives ...|2023|
1941|
| B08WJSHSLC|Fire TV Stick (3rd Gen) with Alexa Voice Remote (includes TV controls) + Star Wars...|2023|
1911|
| B07H65KP63|                                Echo Dot (3rd Gen, 2018 release) - Smart speaker with Alexa - Charcoal|2023|
1723|
| B0C3NMZ2C7|Fire TV Stick 4K, brilliant 4K streaming quality, TV and smart home controls, free...|2023|
1711|
| B0CB5683SY|LaView 4MP Bulb Security Camera 2.4GHz,360° 2K Security Cameras Wireless Outdoor I...|2023|
1666|
| B0BDSM76W4|Fastest WiFi Extender/ Booster | 2023 release Up to 74% Faster | Broader Coverage ...|2023|
1534|
| B0B5GRGB58|Security Cameras Wireless Outdoor, 1080P Battery Powered AI Motion Detection Spotl...|2023|
1420|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
```

```
In [93]: # With Spark SQL
spark.sql("""
SELECT
  parent_asin,
  product_name,
  YEAR(datetime) AS year,
```

```

COUNT(*) AS 5_star_verified_purchase_reviews
FROM joined_view
WHERE
    YEAR(datetime) = 2023
    AND verified_purchase = 'true'
    AND rating = 5.0
GROUP BY parent_asin, product_name, YEAR(datetime)
ORDER BY 5_star_verified_purchase_reviews DESC
LIMIT 10
""").show(truncate = 85)

```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
|parent_asin|                                     product_name|year|5_star_verified_purchase_re
views|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
| B0BZ6XH3LC|ZIUTY Wireless Earbuds, Bluetooth 5.3 Headphones 50H Playtime with LED Digital Dis...|2023|
2976|
| B0BNZ2785V|TAGRY Bluetooth Headphones True Wireless Earbuds 60H Playback LED Power Display Ea...|2023|
2953|
| B0C1ZHJRGB|Wireless Earbuds Bluetooth Headphones 48hrs Play Back Sport Earphones with LED Dis...|2023|
2298|
| B0B4JQY8JY|Sunany USB Flash Drive 256 GB for Phone and Pad, High Speed External Thumb Drives ...|2023|
1941|
| B08WJSHSLC|Fire TV Stick (3rd Gen) with Alexa Voice Remote (includes TV controls) + Star Wars...|2023|
1911|
| B07H65KP63|                               Echo Dot (3rd Gen, 2018 release) - Smart speaker with Alexa - Charcoal|2023|
1723|
| B0C3NMZ2C7|Fire TV Stick 4K, brilliant 4K streaming quality, TV and smart home controls, free...|2023|
1711|
| B0CB5683SY|LaView 4MP Bulb Security Camera 2.4GHz,360° 2K Security Cameras Wireless Outdoor I...|2023|
1666|
| B0BDSM76W4|Fastest WiFi Extender/ Booster | 2023 release Up to 74% Faster | Broader Coverage ...|2023|
1534|
| B0B5GRGB58|Security Cameras Wireless Outdoor, 1080P Battery Powered AI Motion Detection Spotl...|2023|
1420|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+

```

Looking at the product names, small electronic devices were the most popular in 2023. Wireless audio devices (earbuds/headphones) were by far the most popular. Aside from wireless audio devices, other popular electronics in 2023 were flash drives and home improvement products.

Close the SparkSession

```
In [94]: # Stop the SparkSession  
spark.stop()
```

Conclusion

As the data was scraped, there may be some issues with the scraping process, which included books in the data.

Memory cards and bluetooth audio devices are extremely popular. Amazon could try to tap into the demand for these items by developing their own Amazon Basics versions of these.

If Amazon already has their own brand of such products, an investigation into why they aren't selling as well as products by other brands should be conducted. For example, how does sound quality and connectivity compare with other brands? Or how do read and write speeds compare to memory cards from other brands?

Echo Dots and Fire TV Sticks are still popular and should continue to be developed.