Amazon_preprocessing

```
val ratingsDF = spark.read
  .option("inferSchema", "true")
  .option("header", "true")
  .csv("/user/sc10670_nyu_edu/project/*.csv.gz")
```

SPARK JOB FINISHED

```
ratingsDF: org.apache.spark.sql.DataFrame = [user_id: string, parent_asin: string ... 2 more fields]
```

Took 2 min 58 sec. Last updated by anonymous at November 22 2024, 5:07:40 PM.

```
ratingsDF.printSchema()
```

FINISHED

```
root
 |-- user_id: string (nullable = true)
 |-- parent_asin: string (nullable = true)
 |-- rating: double (nullable = true)
 |-- timestamp: long (nullable = true)
```

Took 1 sec. Last updated by anonymous at November 22 2024, 5:07:41 PM.

```
z.show(ratingsDF)
```

SPARK JOB (http://nyu-dataproc-sw-8w9r.c.hpc-dataproc-19b8.internal:38833/jobs/job?id=2) FINISHED

settings

user_id	parent_asin	rating
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B00J10VZ2W	5.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B09KX3FZQS	1.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B0CGY43Y3P	3.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B08K8N5FB2	5.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B00GUAURXY	5.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B0C6V27S6N	5.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B08CSZDXZY	5.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B09G2PW8ZG	2.0

Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult**

Took 0 sec. Last updated by anonymous at November 22 2024, 5:07:41 PM.

```
val newratingsDF = ratingsDF
  .withColumn("source_file", regexp_replace(
    regexp_extract(input_file_name(), "([^/]+$)", 0),
    "\\.csv", ""
  ))
```

FINISHED

```
newratingsDF: org.apache.spark.sql.DataFrame = [user_id: string, parent_asin: string ... 3 more fields]
```

Took 0 sec. Last updated by anonymous at November 22 2024, 5:07:41 PM.

```
newratingsDF.printSchema()
```

FINISHED

```
root
 |-- user_id: string (nullable = true)
 |-- parent_asin: string (nullable = true)
 |-- rating: double (nullable = true)
 |-- timestamp: long (nullable = true)
 |-- source_file: string (nullable = false)
```

Took 1 sec. Last updated by anonymous at November 22 2024, 5:07:42 PM.

```
z.show(newratingsDF)
```

SPARK JOB (http://nyu-dataproc-sw-8w9r.c.hpc-dataproc-19b8.internal:38833/jobs/job?id=3) FINISHED

settings

user_id	parent_asin	rating
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B00J10VZ2W	5.0

AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B09KX3FZQS	1.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B0CGY43Y3P	3.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B08K8N5FB2	5.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B00GUAURXY	5.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B0C6V27S6N	5.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B08CSZDXZY	5.0
AFKZENTNBQ7A7V7UXW5JJI6UGRYQ	B09G2PW8ZG	2.0

Amazon_preprocessing

Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult**

Took 0 sec. Last updated by anonymous at November 22 2024, 5:07:42 PM.

```
val asinSchema = "parent_asin STRING, category STRING"
```

FINISHED

```
asinSchema: String = parent_asin STRING, category STRING
```

Took 0 sec. Last updated by anonymous at November 22 2024, 5:07:42 PM.

```
val asinSchema = "parent_asin STRING, category STRING"

val asinDF = spark.read
  .schema(asinSchema)
  .csv("/user/sc10670_nyu_edu/project/asin.csv")
```

FINISHED

```
asinSchema: String = parent_asin STRING, category STRING
asinDF: org.apache.spark.sql.DataFrame = [parent_asin: string, category: string]
```

Took 1 sec. Last updated by anonymous at November 22 2024, 5:07:43 PM.

```
asinDF.printSchema()
```

FINISHED

```
root
 |-- parent_asin: string (nullable = true)
 |-- category: string (nullable = true)
```

Took 0 sec. Last updated by anonymous at November 22 2024, 5:07:43 PM.

```
z.show(asinDF)
```

SPARK JOB (http://nyu-dataproc-sw-8w9r.c.hpc-dataproc-19b8.internal:38833/jobs/job?id=4) FINISHED

settings

parent_asin	category
B07R3DYMH6	Home and Kitchen
0701169850	Books
B09X1MRDN6	Clothing Shoes and Jewelry
B073C4Q7W8	Clothing Shoes and Jewelry
B01HDXC8AG	Sports and Outdoors
0435088688	Books
0316185361	Books
B08BLDKYHB	Beauty and Personal Care

Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult**

Took 0 sec. Last updated by anonymous at November 22 2024, 5:07:43 PM.

```
println(s"Number of records in the ratings dataframe: ${newratingsDF.count()}")
println(s"Number of records in the asin dataframe: ${asinDF.count()}")
```

SPARK JOB FINISHED

```
Number of records in the ratings dataframe: 107426970
Number of records in the asin dataframe: 35393189
```

Took 46 sec. Last updated by anonymous at November 22 2024, 5:08:29 PM.

```
z.show(newratingsDF.summary())
```

SPARK JOB (http://nyu-dataproc-sw-8w9r.c.hpc-dataproc-19b8.internal:38833/jobs/job?id=9) ABORT

```
org.apache.spark.SparkException: Job 9 cancelled part of cancelled job group zeppelin|sc10670_nyu_edu|2KCRHK9DV|paragraph_1732257082518_120361418
  at org.apache.spark.scheduler.DAGScheduler.failJobAndIndependentStages(DAGScheduler.scala:2844)
  at org.apache.spark.scheduler.DAGScheduler.handleJobCancellation(DAGScheduler.scala:2719)
  at org.apache.spark.scheduler.DAGScheduler.$anonfun$handleJobGroupCancelled$4(DAGScheduler.scala:1193)
  at scala.runtime.java8.JFunction1$mcVI$sp.apply(JFunction1$mcVI$sp.java:23)
  at scala.collection.mutable.HashSet.foreach(HashSet.scala:79)
  at org.apache.spark.scheduler.DAGScheduler.handleJobGroupCancelled(DAGScheduler.scala:1192)
  at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.doOnReceive(DAGScheduler.scala:3004)
  at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:2982)
  at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:2971)
  at org.apache.spark.util.EventLoop$$anon$1.run(EventLoop.scala:49)
```

Took 27 min 34 sec. Last updated by anonymous at November 22 2024, 5:36:03 PM.

Amazon_preprocessing

```
val missingRatingsDF = newratingsDF.filter(newratingsDF.columns.map(c => col(c).isNull).reduce(_ || _))
println(s"Number of records with null values in newratingsDF: ${missingRatingsDF.count()}")
```

SPARK JOB FINISHED

```
Number of records with null values in newratingsDF: 0
missingRatingsDF: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [user_id: string, parent_asin: string ... 3 more fields]
```

Took 1 min 36 sec. Last updated by anonymous at November 22 2024, 5:37:58 PM.

```
val missingAsinDF = asinDF.filter($"parent_asin".isNull || $"category".isNull)
println(s"Number of records with null values in asinDF: ${missingAsinDF.count()}")
```

SPARK JOB FINISHED

```
Number of records with null values in asinDF: 0
missingAsinDF: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [parent_asin: string, category: string]
```

Took 26 sec. Last updated by anonymous at November 22 2024, 5:38:24 PM.

```
val mergedDF = newratingsDF.join(asinDF, Seq("parent_asin"), "left_outer")

val nullData = mergedDF.filter($"category".isNull)

// Optional: Replace null values in `category` if needed
//val filledDF = mergedDF.na.fill(Map("category" -> "Unknown"))

// Display the merged DataFrame or null rows
//filledDF.show()
nullData.show()
```

SPARK JOB FINISHED

```
+-----------+-------+------+---------+-----------+--------+
|parent_asin|user_id|rating|timestamp|source_file|category|
+-----------+-------+------+---------+-----------+--------+
+-----------+-------+------+---------+-----------+--------+

mergedDF: org.apache.spark.sql.DataFrame = [parent_asin: string, user_id: string ... 4 more fields]
nullData: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [parent_asin: string, user_id: string ... 4 more fields]
```

Took 10 min 8 sec. Last updated by anonymous at November 22 2024, 5:48:32 PM.

```
val updatedCategoriesDF = mergedDF.withColumn(
  "category",
  when(col("category").isNull, regexp_replace(col("source_file"), "_", " ")).otherwise(col("category"))
)
```

FINISHED

```
updatedCategoriesDF: org.apache.spark.sql.DataFrame = [parent_asin: string, user_id: string ... 4 more fields]
```

Took 0 sec. Last updated by anonymous at November 22 2024, 5:48:32 PM.

```
val nonRedundantDF = updatedCategoriesDF.drop("user_id", "parent_asin", "source_file")
```

FINISHED

```
nonRedundantDF: org.apache.spark.sql.DataFrame = [rating: double, timestamp: bigint ... 1 more field]
```

Took 0 sec. Last updated by anonymous at November 22 2024, 5:48:32 PM.

```
val cleanedDF = nonRedundantDF.withColumn(
  "date",
  from_unixtime(col("timestamp") / 1000).cast("date")
).drop("timestamp")
```

FINISHED

```
cleanedDF: org.apache.spark.sql.DataFrame = [rating: double, category: string ... 1 more field]
```

Took 0 sec. Last updated by anonymous at November 22 2024, 5:48:32 PM.

Amazon_preprocessing

```
cleanedDF.show()
```

SPARK JOB FINISHED

```
+------+--------------------+----------+
|rating|            category|      date|
+------+--------------------+----------+
|   5.0|Arts Crafts and S...|2015-09-03|
|   5.0|               Books|2018-06-23|
|   5.0|               Books|2018-06-23|
|   5.0|Clothing Shoes an...|2016-01-13|
|   5.0|Beauty and Person...|2013-02-19|
|   5.0|         Electronics|2020-01-09|
|   5.0|Clothing Shoes an...|2012-09-30|
|   5.0|          Automotive|2021-05-20|
|   5.0|Arts Crafts and S...|2020-02-10|
|   5.0|Health and Household|2016-01-04|
|   5.0|          Automotive|2019-10-17|
|   5.0|Beauty and Person...|2020-12-30|
|   5.0|         Electronics|2013-10-24|
|   5.0|Health and Household|2019-03-18|
|   5.0|Health and Household|2020-02-06|
|   3.0|    Home and Kitchen|2019-03-18|
|   5.0|               Books|2018-11-10|
|   5.0|Grocery and Gourm...|2015-09-03|
|   5.0|Beauty and Person...|2016-12-15|
|   5.0|    Home and Kitchen|2020-03-07|
+------+--------------------+----------+
only showing top 20 rows
```

Took 55 sec. Last updated by anonymous at November 22 2024, 5:49:27 PM.

```
val outputPath = "/user/sc10670_nyu_edu/project/amazon-clean.parquet"

cleanedDF.write.mode("overwrite").parquet(outputPath)
```

SPARK JOB FINISHED

```
outputPath: String = /user/sc10670_nyu_edu/project/amazon-clean.parquet
```

Took 10 min 58 sec. Last updated by anonymous at November 22 2024, 6:00:25 PM.

REVIEW ANALYSIS/PROFILING

FINISHED

```
val catCountDF = cleanedDF.groupBy("category").agg(count("category") as "category_count").orderBy(desc("category_count"))
```

```
catCountDF: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [category: string, category_count: bigint]
```

Took 1 sec. Last updated by anonymous at November 22 2024, 6:00:26 PM.

```
z.show(catCountDF)
```

SPARK JOB FINISHED

settings

category	category_count
Home and Kitchen	28202600
Clothing Shoes and Jewelry	23102537
Electronics	15473536
Books	9488297
Health and Household	7176552
Beauty and Personal Care	6624441
Automotive	6072233
Grocery and Gourmet Food	3948741

Took 10 min 5 sec. Last updated by anonymous at November 22 2024, 12:26:01 AM. (outdated)

```
val catRatingsDF = cleanedDF
  .groupBy("category")
  .agg(avg("rating").as("avg_rating"))
  .orderBy(desc("avg_rating"))
```

FINISHED

```
catRatingsDF: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [category: string, avg_rating: double]
```

Took 0 sec. Last updated by anonymous at November 22 2024, 6:00:26 PM.

```
z.show(catRatingsDF)
```

SPARK JOB FINISHED

settings

Amazon_preprocessing

category	avg_rating
Gift Cards	4.913544668587896
CDs and Vinyl	4.482091290112342
Arts Crafts and Sewing	4.477071399663128
Automotive	4.3995568022505065
Books	4.396695845418836
Health and Household	4.348558193405412
Home and Kitchen	4.334334387609653
Baby Products	4.325543094216906

Took 11 min 2 sec. Last updated by anonymous at November 22 2024, 6:11:28 PM. (outdated)

```
val oldestDateDF=cleanedDF.orderBy("date")
```

FINISHED

```
oldestDateDF: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [rating: double, category: string ... 1 more field]
```

Took 0 sec. Last updated by anonymous at November 22 2024, 6:11:28 PM.

```
z.show(oldestDateDF)
```

SPARK JOB FINISHED

settings

rating	category
5.0	Books
4.0	Books
5.0	Books
4.0	Books
5.0	Books
5.0	Books
5.0	Books
5.0	Books

Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult**

Took 11 min 24 sec. Last updated by anonymous at November 22 2024, 6:22:52 PM. (outdated)

READY