AXG180147

Project Part 2:

Introduction and problem description

Initially, it was difficult to find out products having a high response rate as compared to other products. Due to this, it was difficult to decide on which products we should move our marketing focus to improve response and sales. Since customer rating is a very important parameter which can impact the purchase decision of future customers. Hence if some products are getting poor rating then those products need more focus and promotion. To find out these products, I have decided to analyze the amazon reviews database to gather information related to products amazon is offering and ratings those products are getting from customers. After analyzing Amazon Reviews Database, I realized for few product categories we are having very low ratings as compared to other product categories. My analysis involves the below steps:

Data Cleaning

In the Amazon Review database there are many records with multiple reviews by the same users for the same product. It is not appropriate to use this data for analysis, it may cause misinterpretation. Hence it is better to exclude such data from the database by creating a filter_view and excluded records having multiple reviews by the same users for the same product.

```
from pyspark.sql import functions as F
```

```
# Load Data Set
df = spark.read\
          .option("header", "true")\
          .option("inferSchema", "true")\
          .option("basePath", "hdfs:///hive/amazon-reviews-pds/parquet/")\
          .parquet("hdfs:///hive/amazon-reviews-pds/parquet/*")
```

```
df.printSchema()
```

Output:

```
root
 |-- marketplace: string (nullable = true)
 |-- customer_id: string (nullable = true)
 |-- review_id: string (nullable = true)
 |-- product_id: string (nullable = true)
 |-- product_parent: string (nullable = true)
 |-- product_title: string (nullable = true)
 |-- star_rating: integer (nullable = true)
 |-- helpful_votes: integer (nullable = true)
 |-- total_votes: integer (nullable = true)
 |-- vine: string (nullable = true)
 |-- verified_purchase: string (nullable = true)
 |-- review_headline: string (nullable = true)
 |-- review_body: string (nullable = true)
```

```
 |-- review_date: date (nullable = true)
 |-- year: integer (nullable = true)
 |-- product_category: string (nullable = true)
```

```
df.columns
```

Output:

```
['marketplace', 'customer_id', 'review_id', 'product_id', 'product_parent',
'product_title', 'star_rating', 'helpful_votes', 'total_votes', 'vine',
'verified_purchase', 'review_headline', 'review_body', 'review_date', 'year',
'product_category']
```

Filter Dataset:

```
from pyspark.sql.window import *
from pyspark.sql.functions import row_number
x=df.withColumn("row_number",row_number().over(Window.partitionBy("customer_id",
"product_id").orderBy("customer_id","product_id")))
y= x.filter(F.col("year")>2004)
fltr_data = y.row_number.isin(1)
aftr_fltr=y.where(fltr_data)
aftr_fltr.persist()
```

Output:

```
DataFrame[marketplace: string, customer_id: string, review_id: string,
product_id: string, product_parent: string, product_title: string, star_rating:
int, helpful_votes: int, total_votes: int, vine: string, verified_purchase:
string, review_headline: string, review_body: string, review_date: date, year:
int, product_category: string, row_number: int]
```

Basic Exploratory Analysis:

Carried out basic exploratory analysis to understand a basic overview of the Amazon Review Database and calculated different fields like Number of reviews, Number of users, Average and Median review stars, Percentiles of length of the review, Percentiles for number of reviews per product, Identify week number (each year has 52 weeks) for each year and product category with most positive reviews (4 and 5 star).

1.1 Number of reviews

```
aftr_fltr.groupby("year","product_category").agg(F.countDistinct("review_id").al
ias('Number of Reviews')).show(5)
```

Output:

```
+----+--------------------+-----------------+
|year|    product_category|Number of Reviews|
+----+--------------------+-----------------+
|2014|               Books|          3540840|
|2010|Digital_Ebook_Pur...|           102513|
|2015|               Books|          2860736|
|2013|            Wireless|          1767127|
|2014|         Mobile_Apps|          1728286|
+----+--------------------+-----------------+
only showing top 5 rows
```

1.2 Number of users

```
aftr_fltr.groupby("year","product_category").agg(F.countDistinct("customer_id").
alias('Number of Users')).show(5)
```

Output:

```
+----+--------------------+---------------+
|year|    product_category|Number of Users|
+----+--------------------+---------------+
|2014|               Books|        1859223|
|2010|Digital_Ebook_Pur...|          61197|
|2015|               Books|        1548552|
|2013|            Wireless|        1193454|
|2014|         Mobile_Apps|         988656|
+----+--------------------+---------------+
only showing top 5 rows
```

1.3 Average and Median review stars

```
aftr_fltr.groupby("year","product_category").agg(round(F.avg("star_rating"),3).a
lias('Avg_Rating'),

F.expr('percentile_approx(star_rating,0.5)').alias('Median_Rating')).show()
```

Output:

```
+----+--------------------+----------+-------------+
|year|    product_category|Avg_Rating|Median_Rating|
+----+--------------------+----------+-------------+
|2014|               Books|     4.473|            5|
|2000|           Video_DVD|     4.012|            5|
|2010|Digital_Ebook_Pur...|     3.822|            4|
|2003|           Video_DVD|     4.036|            5|
|2015|               Books|     4.497|            5|
|2013|            Wireless|      3.82|            4|
|2014|         Mobile_Apps|     3.969|            5|
|2002|Digital_Ebook_Pur...|     3.667|            5|
|1999|Digital_Ebook_Pur...|       5.0|            5|
```

```
|2013|Digital_Video_Dow...|     4.208|            5|
|1996|           Video_DVD|     4.333|            5|
|2004|Digital_Ebook_Pur...|     4.538|            5|
|2011|Digital_Ebook_Pur...|     4.056|            5|
|2008|               Books|     4.233|            5|
|2001|               Books|     4.197|            5|
|2002|Digital_Video_Dow...|       4.2|            4|
|2012|         Mobile_Apps|     3.995|            5|
|2008|            Wireless|      3.77|            4|
|2011|               Books|     4.251|            5|
|2004|           Video_DVD|     3.989|            5|
+----+--------------------+----------+-------------+
only showing top 20 rows
```

1.4 Percentiles of length of the review.

```
from pyspark.sql.functions import length,count, mean, stddev_pop, min, max
df1=aftr_fltr.withColumn('length', length(df.review_body))
df2=df1.groupby("year","product_category").agg(F.avg("length").alias('avg of Reviews'))
colName = "avg of Reviews"
quantileProbs = [0.1, 0.25, 0.5, 0.75, 0.9, 0.95]
relError = 0.05
df2.stat.approxQuantile("avg of Reviews", quantileProbs, relError)
```

Output:

```
[391.7476834010908, 531.1702520786591, 744.8571428571429, 901.7347037175156, 1091.1026519657087, 3200.6666666666665]
```

1.5 Percentiles for number of reviews per product.

```
from pyspark.sql.functions import length,count, mean, stddev_pop, min, max
df1=aftr_fltr.groupby("year","product_id","product_category").agg(F.countDistinct("review_id").alias('Number of Reviews'))
colName = "Number of Reviews"
quantileProbs = [0.1, 0.25, 0.5, 0.75, 0.9, 0.95]
relError = 0.05
df1.stat.approxQuantile("Number of Reviews", quantileProbs, relError)
```

Output:

```
[1.0, 1.0, 2.0, 5.0, 4428.0, 31128.0]
```

1.6 Identify week number (each year has 52 weeks) for each year and product category with most positive reviews (4 and 5 star)

```
from pyspark.sql.functions import *
X = aftr_fltr.star_rating.isin(4)
Y = aftr_fltr.star_rating.isin(5)
updated_df=aftr_fltr.select("product_category","year","review_date").withColumn(
"week_number",weekofyear("review_date")).where(X | Y)
df_2 =
updated_df.groupby("product_category","year","week_number").agg(F.countDistinct(
"week_number").alias("count"))
df_2.drop('count').show()
```

Output:

```
+--------------------+----+-----------+
|    product_category|year|week_number|
+--------------------+----+-----------+
|Digital_Ebook_Pur...|2014|         11|
|Digital_Ebook_Pur...|2015|         16|
|           Video_DVD|2015|         12|
|           Video_DVD|2011|         37|
|               Books|2011|         36|
|               Books|2007|         37|
|               Books|2008|         48|
|               Books|2004|         18|
|                  PC|2003|         15|
|           Video_DVD|2000|         46|
|           Video_DVD|2000|          9|
|                  PC|2003|         14|
|           Video_DVD|1998|         29|
|               Books|1996|          6|
|            Wireless|1999|         47|
|Digital_Ebook_Pur...|2015|         11|
|                  PC|2012|         24|
|               Books|2014|          6|
|               Books|2010|         12|
|Digital_Ebook_Pur...|2013|         49|
+--------------------+----+-----------+
only showing top 20 rows
```

2. Provide detailed analysis of "Digital eBook Purchase" versus Books.

Performed detailed analysis of "Digital eBook Purchase" versus Books.

2.1. Using Spark Pivot functionality, produce Data Frame with following columns:

Using Spark Pivot functionality, produce Data Frame with following columns:

1. **Year**
2. **Month**
3. **Total number of reviews for "Digital eBook Purchase" category**
4. **Total number of reviews for "Books" category**
5. **Average stars for reviews for "Digital eBook Purchase" category**
6. **Average stars for reviews for "Books" category**

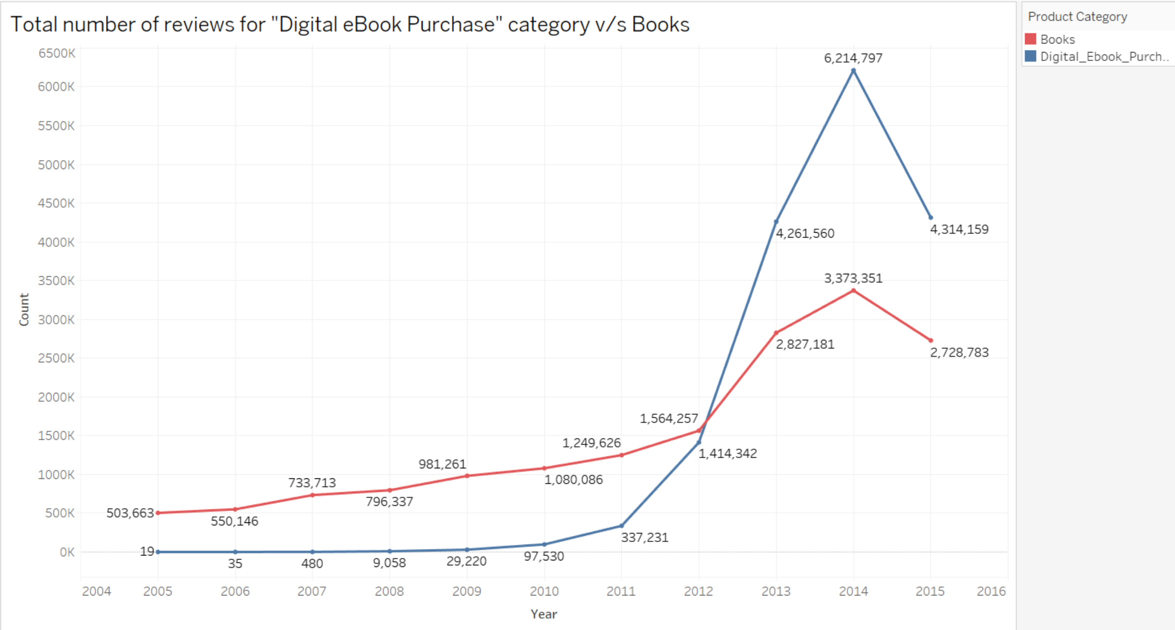
```
to_pivot=['Digital_Ebook_Purchase','Books']
pivoted=aftr_fltr.groupBy("year",F.month(F.col("review_date"))).pivot("product_c
ategory",to_pivot)\
 .agg((F.count("review_id")).alias("count_of_reviews"),
 F.round(F.mean("star_rating"),3).alias("Avg_star_rating")).sort("year","month(r
eview_date)",ascending=True).show()
```

Output:

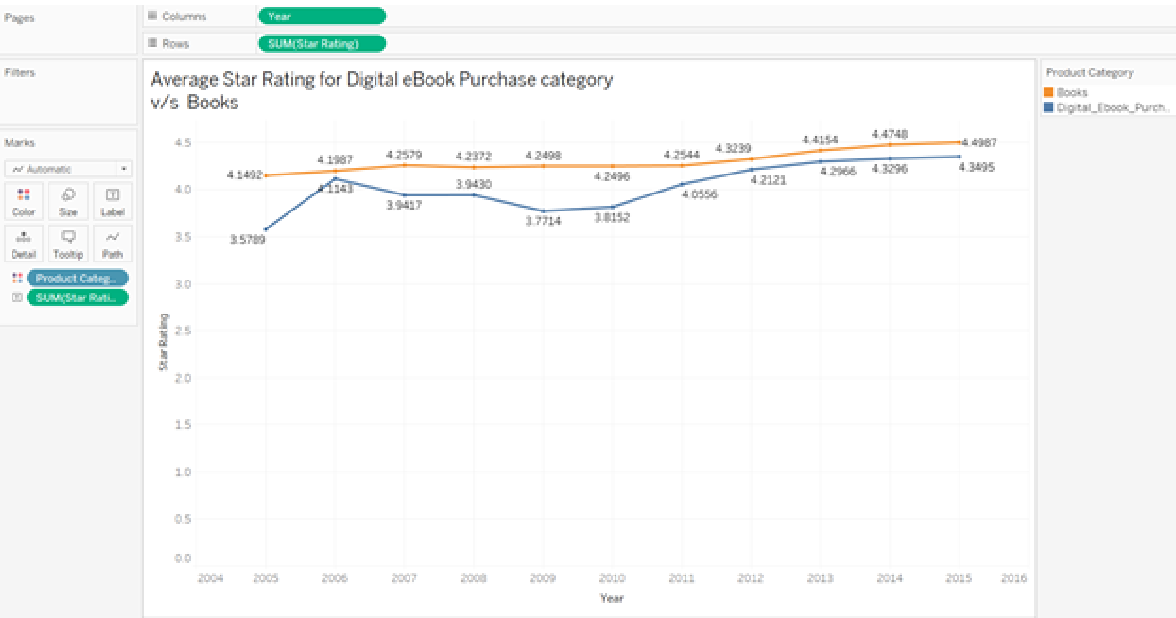
```
+----+------------------+---------------------------------------+---------------
-----------------------+----------------------+---------------------+
|year|month(review_date)|Digital_Ebook_Purchase_count_of_reviews|Digital_Ebook_P
urchase_Avg_star_rating|Books_count_of_reviews|Books_Avg_star_rating|
+----+------------------+---------------------------------------+---------------
-----------------------+----------------------+---------------------+
|2005|                 1|                                      1|
                   5.0|                 40420|                 4.12|
|2005|                 2|                                   null|
                  null|                 33714|                4.124|
|2005|                 3|                                      2|
                   4.5|                 38870|                4.122|
|2005|                 4|                                      1|
                   5.0|                 36881|                4.132|
|2005|                 5|                                      1|
                   1.0|                 36871|                4.132|
|2005|                 6|                                   null|
                  null|                 36605|                4.115|
|2005|                 7|                                      3|
                   2.0|                 45941|                4.128|
|2005|                 8|                                      3|
                 2.667|                 58922|                4.186|
|2005|                 9|                                      2|
                   4.0|                 58129|                4.203|
|2005|                10|                                      4|
                   4.0|                 51210|                 4.18|
|2005|                11|                                      1|
                   5.0|                 40890|                4.151|
|2005|                12|                                      1|
                   5.0|                 42523|                4.126|
|2006|                 1|                                      8|
                 3.375|                 51994|                4.135|
|2006|                 2|                                      5|
                   4.6|                 54415|                4.203|
|2006|                 3|                                   null|
                  null|                 66894|                4.233|
|2006|                 4|                                   null|
                  null|                 27672|                4.132|
|2006|                 5|                                      1|
                   5.0|                 45005|                 4.18|
|2006|                 6|                                      5|
                   4.2|                 48050|                4.184|
|2006|                 7|                                      1|
                   4.0|                 55793|                  4.2|
|2006|                 8|                                      9|
                 4.444|                 54421|                4.213|
```

```
+----+------------------+---------------------------------------+---------------
----------------------+---------------------+--------------------+
only showing top 20 rows
```

2.2 Produce two graphs to demonstrate aggregations



From this graph we can interpret number of reviews for Digital E-book Purchase category are more as compared to Books category.



From this graph we can interpret that average star rating for books category is high as compared to digital e-book purchase.

3. Identify similar products (books) in both categories. Use "product_title" to match products. To account for potential differences in naming of products, compare titles after stripping spaces and converting to lower case.

Performed analysis to Identify similar products (books) in both categories. Use "product_title" to match products.

3.1 Is there a difference in average rating for the similar books in digital and printed form?

```
product_filter=['Digital_Ebook_Purchase']
digital_ebook=aftr_fltr.groupBy("product_title","product_category")\
 .agg((F.count("review_id")).alias("no_of_dig_book_reviews"),
 F.round(F.mean("star_rating"),3).alias("Avg_rating_of_dig_book")).filter(F.col(
"product_category").isin(product_filter))

trimmed_dig_book=digital_ebook.select(F.lower(F.trim(F.col("product_title"))).al
ias("product_title"),F.col("no_of_dig_book_reviews") \
                      ,F.col("Avg_rating_of_dig_book"))

variable=['Books']
books=aftr_fltr.groupBy("product_title","product_category")\
 .agg((F.count("review_id")).alias("no_of_reviews_for_book"),
 F.round(F.mean("star_rating"),3).alias("Avg_rating_of_book")).filter(F.col("pro
duct_category").isin(variable))

trimmed_book=books.select(F.lower(F.trim(F.col("product_title"))).alias("product
_title"),F.col("no_of_reviews_for_book") \
                      ,F.col("Avg_rating_of_book"))

joinExpression = trimmed_book["product_title"] ==
trimmed_dig_book["product_title"]
joinType = "inner"
out=trimmed_book.join(trimmed_dig_book, joinExpression, joinType)

out.show()
```

Output:

```
+--------------------+----------------------+------------------+----------------
----+----------------------+----------------------+
|       product_title|no_of_reviews_for_book|Avg_rating_of_book|
product_title|no_of_dig_book_reviews|Avg_rating_of_dig_book|
+--------------------+----------------------+------------------+----------------
----+----------------------+----------------------+
|"rays of light": ...|                     2|               5.0|"rays of light":
...|                     1|                   5.0|
|"the siege of khe...|                    19|             4.316|"the siege of
khe...|                   156|                 3.327|
|           'dem bon'z|                     4|               5.0|           'dem
bon'z|                     2|                   5.0|
|    0400 roswell time|                     1|               5.0|    0400 roswell
time|                     6|                 3.667|
|10 smart things g...|                    19|             4.789|10 smart things
g...|                     6|                 4.833|
|10 smart things g...|                     1|               5.0|10 smart things
g...|                     6|                 4.833|
|100 prayers for y...|                    11|               5.0|100 prayers for
y...|                     7|                   5.0|
```

```
|13 cent killers: ...|                    37|             2.811|13 cent killers:
...|                     15|                  3.933|
|25 essentials: te...|                    41|             4.439|25 essentials:
te...|                      1|                    5.0|
|30 before 30: tra...|                     2|               3.5|30 before 30:
tra...|                     33|                    4.97|
|300 hard word sea...|                     2|               4.5|300 hard word
sea...|                      7|                    1.0|
|42 rules to incre...|                     1|               5.0|42 rules to
incre...|                      2|                    5.0|
|50 american heroe...|                     2|               5.0|50 american
heroe...|                      3|                    4.0|
|50 successful har...|                    49|             4.347|50 successful
har...|                      3|                  4.667|
|52 prepper projec...|                    30|               3.9|52 prepper
projec...|                      2|                    4.5|
|73 north: the bat...|                     6|               5.0|73 north: the
bat...|                      1|                    5.0|
|<i>change</i> the...|                     8|              4.75|<i>change</i>
the...|                      2|                    4.5|
|      a changed life|                     5|               4.2|      a changed
life|                    36|                  4.222|
|a chip off the ol...|                     1|               5.0|a chip off the
ol...|                      1|                    5.0|
|a closer look at ...|                     1|               5.0|a closer look at
...|                     1|                    1.0|
+--------------------+----------------------+------------------+----------------
----+----------------------+----------------------+
only showing top 20 rows
```

3.2 Calculate number of items with high stars in digital form versus printed form, and vise versa. Alternatively, you can make the conclusion by using appropriate pairwise statistic.

```
star_rating=F.col("Avg_rating_of_book")>4
out.where(star_rating).count()
star_rating1=F.col("Avg_rating_of_dig_book")>4
out.where(star_rating1).count()
```

Output:

```
245528
```

From output we can interpret that printed book has got more number of higher rating. Hence count of more than 4 star ratings is higher for printed books as compared to digital book star ratings.

4. Using provided LDA starter notebook, perform LDA topic modeling for the reviews in Digital_Ebook_Purchase and Books categories. Consider reviews for the January of 2015 only.

Performed LDA to analyze effectiveness of topic modelling in both star rating 4/5and 1/2 cases.

```
from pyspark.mllib.clustering import LDA, LDAModel
from pyspark.mllib.linalg import Vectors
from pyspark.ml.feature import CountVectorizer, IDF,RegexTokenizer, Tokenizer
from pyspark.sql.types import ArrayType
from pyspark.sql.types import StringType
from pyspark.sql.types import *
from pyspark.sql.functions import udf
from pyspark.sql.functions import struct
import re
from pyspark.ml.feature import StopWordsRemover
from pyspark.ml.clustering import LDA
from pyspark.ml.feature import CountVectorizer
```

4.2 Stop words

```
stop_words = ['a', 'about', 'above', 'across', 'after', 'afterwards', 'again',
'against', 'all', 'almost', 'alone', 'along', 'already', 'also', 'although',
'always', 'am', 'among', 'amongst', 'amoungst', 'amount', 'an', 'and',
'another', 'any', 'anyhow', 'anyone', 'anything', 'anyway', 'anywhere', 'are',
'around', 'as', 'at', 'back', 'be', 'became', 'because', 'become', 'becomes',
'becoming', 'been', 'before', 'beforehand', 'behind', 'being', 'below',
'beside', 'besides', 'between', 'beyond', 'bill', 'both', 'bottom', 'but', 'by',
'call', 'can', 'cannot', 'cant', 'co', 'computer', 'con', 'could', 'couldnt',
'cry', 'de', 'describe', 'detail', 'do', 'done', 'down', 'due', 'during',
'each', 'eg', 'eight', 'either', 'eleven', 'else', 'elsewhere', 'empty',
'enough', 'etc', 'even', 'ever', 'every', 'everyone', 'everything',
'everywhere', 'except', 'few', 'fifteen', 'fify', 'fill', 'find', 'fire',
'first', 'five', 'for', 'former', 'formerly', 'forty', 'found', 'four', 'from',
'front', 'full', 'further', 'get', 'give', 'go', 'had', 'has', 'hasnt', 'have',
'he', 'hence', 'her', 'here', 'hereafter', 'hereby', 'herein', 'hereupon',
'hers', 'herself', 'him', 'himself', 'his', 'how', 'however', 'hundred', 'i',
'ie', 'if', 'in', 'inc', 'indeed', 'interest', 'into', 'is', 'it', 'its',
'itself', 'keep', 'last', 'latter', 'latterly', 'least', 'less', 'ltd', 'made',
'many', 'may', 'me', 'meanwhile', 'might', 'mill', 'mine', 'more', 'moreover',
'most', 'mostly', 'move', 'much', 'must', 'my', 'myself', 'name', 'namely',
'neither', 'never', 'nevertheless', 'next', 'nine', 'no', 'nobody', 'none',
'noone', 'nor', 'not', 'nothing', 'now', 'nowhere', 'of', 'off', 'often', 'on',
'once', 'one', 'only', 'onto', 'or', 'other', 'others', 'otherwise', 'our',
'ours', 'ourselves', 'out', 'over', 'own', 'part', 'per', 'perhaps', 'please',
'put', 'rather', 're', 'same', 'see', 'seem', 'seemed', 'seeming', 'seems',
'serious', 'several', 'she', 'should', 'show', 'side', 'since', 'sincere',
'six', 'sixty', 'so', 'some', 'somehow', 'someone', 'something', 'sometime',
'sometimes', 'somewhere', 'still', 'such', 'system', 'take', 'ten', 'than',
'that', 'the', 'their', 'them', 'themselves', 'then', 'thence', 'there',
'thereafter', 'thereby', 'therefore', 'therein', 'thereupon', 'these', 'they',
'thick', 'thin', 'third', 'this', 'those', 'though', 'three', 'through',
'throughout', 'thru', 'thus', 'to', 'together', 'too', 'top', 'toward',
'towards', 'twelve', 'twenty', 'two', 'un', 'under', 'until', 'up', 'upon',
'us', 'very', 'via', 'was', 'we', 'well', 'were', 'what', 'whatever', 'when',
'whence', 'whenever', 'where', 'whereafter', 'whereas', 'whereby', 'wherein',
'whereupon', 'wherever', 'whether', 'which', 'while', 'whither', 'who',
'whoever', 'whole', 'whom', 'whose', 'why', 'will', 'with', 'within', 'without',
'would', 'yet', 'you', 'your', 'yours', 'yourself', 'yourselves', '']
stop_words = stop_words + ['br','book','34','m','y','zu','ich']
```

4.1 LDA for for reviews with 4/5 stars

```
df_ml = aftr_fltr.filter((F.col("product_category")=="Digital_Ebook_Purchase") \
                    & (F.col("year")==2015) \
                    & (F.col("review_date")<'2015-02-01')
                    & (F.col("star_rating")>3))
df_ml.show()
```

Output:

```
+-----------+----------+-------------+----------+-------------+-------------
------+-----------+-------------+-----------+----+-----------------+------------
--------+-------------------+-----------+----+-------------------+----------+
```

```
|marketplace|customer_id|     review_id|product_id|product_parent|
product_title|star_rating|helpful_votes|total_votes|vine|verified_purchase|
review_headline|         review_body|review_date|year|
product_category|row_number|
+-----------+-----------+--------------+----------+--------------+-------------
------+-----------+-------------+-----------+----+-----------------+------------
--------+--------------------+-----------+----+--------------------+----------+
|         US|   10008277|R2ZUCDNCGQHZC5|B00P8F4NDW|     342710975|CULPABLE
(Spanish...|          5|            0|          0|   N|                Y|
impactante!!!!!|Que se puede deci...| 2015-01-25|2015|Digital_Ebook_Pur...|
    1|
|         US|   10015086|R2TWT8BQB3GYAZ|B00GEEB6X6|     808362562|Forty Acres: A
Th...|          4|            0|          0|   N|                Y|
Riveting !|The book made me ...| 2015-01-06|2015|Digital_Ebook_Pur...|
1|
|         US|   10024344|R1YCZAPG20VY8O|B004ZZS4CC|     559187375|A Stolen Life:
A ...|          5|            0|          0|   N|                Y|
Unbelievably sad.|I felt like I was...| 2015-01-07|2015|Digital_Ebook_Pur...|
      1|
|         US|   10028931|R35OCHI6H7XK5F|B00DXOYUD8|      63212231|Can't Take My
Eye...|          4|            0|          0|   N|                Y|This was a
fun re...|This was a fun re...| 2015-01-29|2015|Digital_Ebook_Pur...|         1|
|         US|   10072578|R2RXKIL3C00L6W|B00S8HLIAI|     908871567|Beautifully
Insig...|          5|            0|          0|   N|                Y|
  AMAZING|I loved loved thi...| 2015-01-15|2015|Digital_Ebook_Pur...|         1|
|         US|   10088522| RSRCRRUW2XWPI|B00PKQ4E7Y|     391201034|Forever
Christmas...|          4|            1|          1|   N|
Y|Forever Christmas...|A lovely collecti...| 2015-01-
07|2015|Digital_Ebook_Pur...|         1|
|         US|   10089203|R2JG9OHU29P5SL|B00B0CR0O6|     659516630|     The Kite
Runner|          5|            0|          0|   N|                Y|Touching,
heart-felt|This book took me...| 2015-01-07|2015|Digital_Ebook_Pur...|
1|
|         US|   10099959|R1X2GC2K98HBSO|B00RQL6A5K|     416690460|   If Not for
Love 2|          5|            0|          0|   N|                N|All I can
say is ...|All I can say is ...| 2015-01-07|2015|Digital_Ebook_Pur...|
1|
|         US|    1014557|R2AA7RMKN63WSK|B009Y3ON4I|     646097776|Hollow City:
The ...|          5|            0|          0|   N|                Y|Hollow City
was a...|Hollow City was a...| 2015-01-12|2015|Digital_Ebook_Pur...|         1|
|         US|   10148387|R1QQNVMN4TP4YS|B00OX553QY|     942873666|Climax: The
Publi...|          5|            1|          1|   N|                Y|
brilliant|As with books 1 a...| 2015-01-29|2015|Digital_Ebook_Pur...|         1|
|         US|   10219710|R1M8OAF56S0YSN|B00KA0AQP4|     884803725|No Prince
Charmin...|          5|            2|          2|   N|                Y|Killian
Stone.......|Killian Stone, CE...| 2015-01-26|2015|Digital_Ebook_Pur...|
1|
|         US|   10278693|R2Q729EBDRXDZC|B00N1ZLI7K|     296851728|When Aliens
Weep:...|          5|            1|          1|   N|                Y|Accinni
for writi...|Thank You to the ...| 2015-01-23|2015|Digital_Ebook_Pur...|
1|
|         US|   10281352| RNFH1TSTVBC39|B00Q13RMVU|     644215742|   A Shade of
Kiev 2|          5|            0|          0|   N|                Y|        A
must read!|Loved the second ...| 2015-01-07|2015|Digital_Ebook_Pur...|
1|
```

```
|         US|   10289050| RAPI752F0MGT6|B003L77W1Y|     923330575|Among the
Brave (...|          5|            0|          1|   N|                Y|
       Good|This book was rea...| 2015-01-20|2015|Digital_Ebook_Pur...|
1|
|         US|    1030479|R1Y264QGYUCYQC|B00BFLDUHS|     445627346|The Lake (The
Lak...|          5|            1|          1|   N|                Y|
The lake|I chose this 5 st...| 2015-01-20|2015|Digital_Ebook_Pur...|         1|
|         US|    1030926|R1DPWBYCILBOSS|B00MTWGMRC|     371274942|Forex: Trading
Su...|          5|            1|          1|   N|                Y|A very
complete b...|I found this book...| 2015-01-11|2015|Digital_Ebook_Pur...|
  1|
|         US|   10319233|R3MJ3OZFJYU71T|B00RXKXYBW|     679650675|Flat Belly
Fruit ...|          5|            1|          2|   N|                Y|
Thank you.|I was not sure ho...| 2015-01-25|2015|Digital_Ebook_Pur...|
1|
|         US|   10367192|R3RUSWNU5WE0K9|B00GUE7B84|     991606429|Live Inspired
Now...|          5|            1|          1|   N|                N|Great
advice! Gre...|Heather is truly ...| 2015-01-07|2015|Digital_Ebook_Pur...|
  1|
|         US|   10442292|R113WJHHWPKK5J|B00PQPJMWG|     342873493|Billionaire
Broth...|          5|            1|          1|   N|                Y|I love a
Dominant...|This book was hot...| 2015-01-09|2015|Digital_Ebook_Pur...|
1|
|         US|   10446142|R2F5MFWNXTL5FL|B004UJISMY|     563840228|The Adventures
of...|          5|            0|          0|   N|                Y|
Five Stars|great book, what ...| 2015-01-09|2015|Digital_Ebook_Pur...|
1|
+-----------+-----------+--------------+----------+--------------+--------------
------+-----------+-------------+-----------+----+-----------------+------------
--------+--------------------+-----------+----+--------------------+----------+
only showing top 20 rows
```

```
df1 = df_ml.withColumn('review_text',
                       F.concat(F.col('review_headline'),F.lit(' '),
F.col('review_body')))
corpus =df1.select('review_text')
```

```
corpus_df = corpus.withColumn("id", F.monotonically_increasing_id())
```

```
corpus_df = corpus_df.dropna()
```

```
corpus_df.persist()
print('Corpus size:', corpus_df.count())
corpus_df.show(5)
```

Output:

```
Corpus size: 437538
+--------------------+---+
|         review_text| id|
+--------------------+---+
|impactante!!!!! Q...|  0|
|Riveting ! The bo...|  1|
|Unbelievably sad....|  2|
|This was a fun re...|  3|
|AMAZING I loved l...|  4|
+--------------------+---+
only showing top 5 rows
```

```
tokenizer = Tokenizer(inputCol="review_text", outputCol="words")
countTokens = udf(lambda words: len(words), IntegerType())
'''
tokenized_df = tokenizer.transform(corpus_df)
tokenized_df.select("review_text", "words").withColumn("tokens",
countTokens(col("words"))).show()
'''
regexTokenizer = RegexTokenizer(inputCol="review_text",
                                outputCol="words",pattern="\\w+", gaps=False)
# alternatively, pattern="\\w+", gaps(False) pattern="\\W"

tokenized_df = regexTokenizer.transform(corpus_df)
tokenized_df.select("review_text", "words") \
    .withColumn("tokens", countTokens(F.col("words"))).show()
```

Output:

```
+--------------------+--------------------+------+
|         review_text|               words|tokens|
+--------------------+--------------------+------+
|impactante!!!!! Q...|[impactante, que,...|    44|
|Riveting ! The bo...|[riveting, the, b...|    21|
|Unbelievably sad....|[unbelievably, sa...|    88|
|This was a fun re...|[this, was, a, fu...|    25|
|AMAZING I loved l...|[amazing, i, love...|   109|
|Forever Christmas...|[forever, christm...|    73|
|Touching, heart-f...|[touching, heart,...|    23|
|All I can say is ...|[all, i, can, say...|    47|
|Hollow City was a...|[hollow, city, wa...|   159|
|brilliant As with...|[brilliant, as, w...|    40|
|Killian Stone.......|[killian, stone, ...|   139|
|Accinni for writi...|[accinni, for, wr...|    58|
|A must read! Love...|[a, must, read, l...|    13|
|Good This book wa...|[good, this, book...|    29|
|The lake I chose ...|[the, lake, i, ch...|    44|
|A very complete b...|[a, very, complet...|    62|
|Thank you. I was ...|[thank, you, i, w...|    32|
|Great advice! Gre...|[great, advice, g...|   241|
|I love a Dominant...|[i, love, a, domi...|    98|
|Five Stars great ...|[five, stars, gre...|    16|
+--------------------+--------------------+------+
only showing top 20 rows
```

```
remover = StopWordsRemover(inputCol="words", outputCol="filtered")
tokenized_df1 = remover.transform(tokenized_df)
tokenized_df1.show(5)
stopwordList = stop_words
```

Output:

```
+--------------------+---+--------------------+--------------------+
|         review_text| id|               words|            filtered|
+--------------------+---+--------------------+--------------------+
|impactante!!!!! Q...|  0|[impactante, que,...|[impactante, que,...|
|Riveting ! The bo...|  1|[riveting, the, b...|[riveting, book, ...|
|Unbelievably sad....|  2|[unbelievably, sa...|[unbelievably, sa...|
|This was a fun re...|  3|[this, was, a, fu...|[fun, read, read,...|
|AMAZING I loved l...|  4|[amazing, i, love...|[amazing, loved, ...|
+--------------------+---+--------------------+--------------------+
only showing top 5 rows
```

```
remover=StopWordsRemover(inputCol="filtered", outputCol="filtered_more"
,stopWords=stopwordList)
tokenized_df2 = remover.transform(tokenized_df1)
tokenized_df2.show(5)
```

Output:

```
+--------------------+---+--------------------+--------------------+------------
--------+
|         review_text| id|               words|            filtered|
filtered_more|
+--------------------+---+--------------------+--------------------+------------
--------+
|impactante!!!!! Q...|  0|[impactante, que,...|[impactante, que,...|[impactante,
que,...|
|Riveting ! The bo...|  1|[riveting, the, b...|[riveting, book, ...|[riveting,
travel...|
|Unbelievably sad....|  2|[unbelievably, sa...|[unbelievably, sa...|
[unbelievably, sa...|
|This was a fun re...|  3|[this, was, a, fu...|[fun, read, read,...|[fun, read,
read,...|
|AMAZING I loved l...|  4|[amazing, i, love...|[amazing, loved, ...|[amazing,
loved, ...|
+--------------------+---+--------------------+--------------------+------------
--------+
only showing top 5 rows
```

```
cv = CountVectorizer(inputCol="filtered_more", outputCol="features", vocabSize =
10000)
cvmodel = cv.fit(tokenized_df2)
featurized_df = cvmodel.transform(tokenized_df2)
vocab = cvmodel.vocabulary
featurized_df.select('filtered_more','features','id').show(5)

countVectors = featurized_df.select('features','id')
countVectors.persist()
print('Records in the DF:', countVectors.count())
```

Output:

```
+--------------------+--------------------+---+
|       filtered_more|            features| id|
+--------------------+--------------------+---+
|[impactante, que,...|(10000,[517,598,7...|  0|
|[riveting, travel...|(10000,[15,54,122...|  1|
|[unbelievably, sa...|(10000,[1,6,11,13...|  2|
|[fun, read, read,...|(10000,[0,34],[4....|  3|
|[amazing, loved, ...|(10000,[0,1,3,8,1...|  4|
+--------------------+--------------------+---+
only showing top 5 rows

Records in the DF: 437538
```

```
lda = LDA(k=10, maxIter=10)
model = lda.fit(countVectors)
```

4.3 Identify 5 top topics for 4/5 rating

```
topics = model.describeTopics(5)
topics_rdd = topics.rdd

topics_words = topics_rdd\
       .map(lambda row: row['termIndices'])\
       .map(lambda idx_list: [vocab[idx] for idx in idx_list])\
       .collect()

for idx, topic in enumerate(topics_words):
    print ("topic: ", idx)
    print ("----------")
    for word in topic:
       print (word)
    print ("----------")
```

Output:

```
topic:  0
----------
story
love
like
read
```

```
loved
----------
topic:  1
----------
read
love
story
great
series
----------
topic:  2
----------
read
series
like
books
story
----------
topic:  3
----------
read
story
good
enjoyed
characters
----------
topic:  4
----------
books
stars
loved
series
read
----------
topic:  5
----------
stars
great
read
excellent
written
----------
topic:  6
----------
good
read
stars
easy
great
----------
topic:  7
----------
great
life
read
god
like
----------
```

```
topic:  8
----------
story
good
really
read
characters
----------
topic:  9
----------
read
story
characters
love
life
----------
```

4.1 LDA for for reviews with 1/2 stars

```
df_ml_2 = aftr_fltr.filter((F.col("product_category")=="Digital_Ebook_Purchase")
\
                   & (F.col("year")==2015) \
                   & (F.col("review_date")<'2015-02-01')
                   & (F.col("star_rating")<3))
df_ml_2.show()
```

Output:

```
+-----------+-----------+--------------+----------+--------------+--------------
------+-----------+-------------+-----------+----+-----------------+------------
--------+-------------------+-----------+----+-------------------+----------+
|marketplace|customer_id|     review_id|product_id|product_parent|
product_title|star_rating|helpful_votes|total_votes|vine|verified_purchase|
review_headline|         review_body|review_date|year|
product_category|row_number|
+-----------+-----------+--------------+----------+--------------+--------------
------+-----------+-------------+-----------+----+-----------------+------------
--------+-------------------+-----------+----+-------------------+----------+
|         US|   10021441|R2LMT4ZCKA88H2|B00P87FLV8|     539739242|Dutch Oven
Cookbo...|          2|            0|          0|   N|                N|
        Meh|Just Ok. 20 recip...| 2015-01-10|2015|Digital_Ebook_Pur...|
1|
|         US|   10457809|R20MSUNE6BNL1J|B005EOAVF6|     978039490|Homeschooling
Boy...|          1|            9|         10|   N|                Y|This was a
waste ...|This was a waste ...| 2015-01-01|2015|Digital_Ebook_Pur...|         1|
|         US|   10459702| R5DAM89L16UHR|B0032UPUOQ|     595863638|Surrender
(Bound ...|          2|            0|          0|   N|
Y|Surrender (Bound ...|This story was no...| 2015-01-
05|2015|Digital_Ebook_Pur...|         1|
|         US|   11311373|R1083ZV1ECOQVP|B00JCJ9X3A|     417649300|Into The Abyss
(D...|          1|            0|         11|   N|                N|Profanity
that ad...|Properly titled. ...| 2015-01-26|2015|Digital_Ebook_Pur...|
1|
```

```
|         US|   11425122| R97MKYNKCMUW5|B00ONDNQYW|      64509529|
Maude|          1|            3|          7|   N|                N|
    Sad|I couldn't connec...| 2015-01-10|2015|Digital_Ebook_Pur...|         1|
|         US|    1145426|R3O1RCCANRBCNB|B00J9RE9HA|     758262835|Muscle
Building: ...|          1|            1|          1|   N|                Y|
     Bad Choice|Very low quality ...| 2015-01-27|2015|Digital_Ebook_Pur...|
    1|
|         US|   11551544|R27JOZ4E336OFJ|B00GVOBMPK|     565713485|Self Help
Masters...|          2|            0|          1|   N|                Y|
       H.mm|I found these \\"...| 2015-01-27|2015|Digital_Ebook_Pur...|
1|
|         US|    1161132| RM8MYL6SEJEM5|B00SS9NLCY|     543984077|HUGE Tatas!
(Pict...|          2|           11|         14|   N|                Y|
Not naked|I'm sorry I like ...| 2015-01-29|2015|Digital_Ebook_Pur...|         1|
|         US|   11712430| RKTW2NMM2LDCB|B004PYDGBM|     448118744|Carrots: A
Shelby...|          1|            1|          9|   N|                Y|
   One Star|Don't bother. Tre...| 2015-01-02|2015|Digital_Ebook_Pur...|
1|
|         US|   11771613|R1V9KNH64IGXAD|B00JMRWJ10|     871728728|Veronica Mars
- t...|          2|            0|          0|   N|                N|
disappointing|the plot of this ...| 2015-01-12|2015|Digital_Ebook_Pur...|
  1|
|         US|    1186783|R1TR22K3JACBHS|B00MMMCQTW|     733371472|The House
Where E...|          2|            4|          4|   N|
Y|Interesting, but ...|I have always bee...| 2015-01-
18|2015|Digital_Ebook_Pur...|         1|
|         US|   12003779|R3G608853TTTQX|B007SGM084|     600633062|Fifty Shades
Tril...|          1|            0|          0|   N|                Y|
One Star|Smutty without an...| 2015-01-03|2015|Digital_Ebook_Pur...|         1|
|         US|   12590424| RZ5FDDMLF93MU|B00IX3FF2O|     359082724|     City of
the Sun|          1|            1|          7|   N|                N|Foul
language and...|About a third or ...| 2015-01-22|2015|Digital_Ebook_Pur...|
    1|
|         US|   12913401|R3ECAQ3BUMHKLR|B006LSZECO|      93816562|  Gone Girl: A
Novel|          1|            0|          2|   N|                N|
Dissapointing|I found the E boo...| 2015-01-21|2015|Digital_Ebook_Pur...|
  1|
|         US|   13097197|R2CPNZRUIVB0VL|B0037471U8|     957923582|An Echo in the
Bo...|          2|            0|          0|   N|                Y|
Two Stars|Not as good as th...| 2015-01-08|2015|Digital_Ebook_Pur...|         1|
|         US|   13124028|R16Z02NQFEZ02Q|B00LEX1N1C|     819566781|The Secrets of
Si...|          2|           11|         17|   N|                Y|Manipulative
and ...|I feel that I am ...| 2015-01-30|2015|Digital_Ebook_Pur...|         1|
|         US|   13170994|R3PG9VVPEVNYG7|B00S3V18KO|     880004017|Royal Kitchen:
me...|          1|           13|         14|   N|                N|    A book of
deceit|This book is not ...| 2015-01-15|2015|Digital_Ebook_Pur...|         1|
|         US|   13181159| RD0U6FZ0CPI91|B000W9164S|     944080987|Serpent's
Tooth: ...|          1|            0|          1|   N|                N|... are
women and...|some of the most ...| 2015-01-25|2015|Digital_Ebook_Pur...|
1|
|         US|   13184825|R1Z6YSTBV5743L|B00HZLCINE|     148365319|Into the
Darkness...|          1|            1|          4|   N|                Y|
        Ewwww|All I can say is ...| 2015-01-19|2015|Digital_Ebook_Pur...|
  1|
|         US|   13698159| R56TYT6WBD54Q|B00LTBWMJ6|     679693886|    Miramont's
Ghost|          1|            0|          1|   N|                Y|
Painful.|I'll admit, I'm o...| 2015-01-30|2015|Digital_Ebook_Pur...|         1|
```

```
+-----------+----------+-------------+---------+-------------+-------------
------+----------+------------+----------+----+----------------+-----------
--------+------------------+----------+----+------------------+----------+
only showing top 20 rows
```

```
df1 = df_ml_2.withColumn('review_text',
                       F.concat(F.col('review_headline'),F.lit(' '),
F.col('review_body')))
corpus =df1.select('review_text')
```

```
corpus_df = corpus.withColumn("id", F.monotonically_increasing_id())
```

```
corpus_df = corpus_df.dropna()
corpus_df.persist()
print('Corpus size:', corpus_df.count())
corpus_df.show(5)
```

Output:

```
Corpus size: 38894
+--------------------+---+
|         review_text| id|
+--------------------+---+
|Meh Just Ok. 20 r...|  0|
|This was a waste ...|  1|
|Surrender (Bound ...|  2|
|Profanity that ad...|  3|
|Sad I couldn't co...|  4|
+--------------------+---+
only showing top 5 rows
```

```
tokenizer = Tokenizer(inputCol="review_text", outputCol="words")
countTokens = udf(lambda words: len(words), IntegerType())
```

```
tokenized_df = tokenizer.transform(corpus_df)
tokenized_df.select("review_text", "words").withColumn("tokens",
countTokens(F.col("words"))).show()
```

Output:

```
+--------------------+--------------------+------+
|         review_text|               words|tokens|
+--------------------+--------------------+------+
|Meh Just Ok. 20 r...|[meh, just, ok., ...|    22|
|This was a waste ...|[this, was, a, wa...|   182|
|Surrender (Bound ...|[surrender, (boun...|   197|
|Profanity that ad...|[profanity, that,...|    80|
```

```
|Sad I couldn't co...|[sad, i, couldn't...|   131|
|Bad Choice Very l...|[bad, choice, ver...|    50|
|H.mm I found thes...|[h.mm, i, found, ...|    37|
|Not naked I'm sor...|[not, naked, i'm,...|    25|
|One Star Don't bo...|[one, star, don't...|     9|
|disappointing the...|[disappointing, t...|    57|
|Interesting, but ...|[interesting,, bu...|   154|
|One Star Smutty w...|[one, star, smutt...|     9|
|Foul language and...|[foul, language, ...|    43|
|Dissapointing I f...|[dissapointing, i...|    73|
|Two Stars Not as ...|[two, stars, not,...|    13|
|Manipulative and ...|[manipulative, an...|   232|
|A book of deceit ...|[a, book, of, dec...|    53|
|... are women and...|[..., are, women,...|    86|
|Ewwww All I can s...|[ewwww, all, i, c...|    30|
|Painful. I'll adm...|[painful., i'll, ...|   100|
+--------------------+--------------------+------+
only showing top 20 rows
```

```
regexTokenizer = RegexTokenizer(inputCol="review_text",
                                outputCol="words",pattern="\\w+", gaps=False)
```

```
tokenized_df = regexTokenizer.transform(corpus_df)
tokenized_df.select("review_text", "words") \
    .withColumn("tokens", countTokens(F.col("words"))).show()
```

Output:

```
+--------------------+--------------------+------+
|         review_text|               words|tokens|
+--------------------+--------------------+------+
|Meh Just Ok. 20 r...|[meh, just, ok, 2...|    21|
|This was a waste ...|[this, was, a, wa...|   191|
|Surrender (Bound ...|[surrender, bound...|   198|
|Profanity that ad...|[profanity, that,...|    76|
|Sad I couldn't co...|[sad, i, couldn, ...|   120|
|Bad Choice Very l...|[bad, choice, ver...|    53|
|H.mm I found thes...|[h, mm, i, found,...|    37|
|Not naked I'm sor...|[not, naked, i, m...|    28|
|One Star Don't bo...|[one, star, don, ...|    11|
|disappointing the...|[disappointing, t...|    57|
|Interesting, but ...|[interesting, but...|   160|
|One Star Smutty w...|[one, star, smutt...|     9|
|Foul language and...|[foul, language, ...|    43|
|Dissapointing I f...|[dissapointing, i...|    71|
|Two Stars Not as ...|[two, stars, not,...|    13|
|Manipulative and ...|[manipulative, an...|   228|
|A book of deceit ...|[a, book, of, dec...|    50|
|... are women and...|[are, women, and,...|    87|
|Ewwww All I can s...|[ewwww, all, i, c...|    31|
|Painful. I'll adm...|[painful, i, ll, ...|   108|
+--------------------+--------------------+------+
only showing top 20 rows
```

```
remover = StopWordsRemover(inputCol="words", outputCol="filtered")
tokenized_df1 = remover.transform(tokenized_df)
tokenized_df1.show(5)

stopwordList = stop_words

remover=StopWordsRemover(inputCol="filtered", outputCol="filtered_more"
,stopWords=stopwordList)
tokenized_df2 = remover.transform(tokenized_df1)
tokenized_df2.show(5)
```

Output:

```
+--------------------+---+--------------------+--------------------+
|         review_text| id|               words|            filtered|
+--------------------+---+--------------------+--------------------+
|Meh Just Ok. 20 r...|  0|[meh, just, ok, 2...|[meh, ok, 20, rec...|
|This was a waste ...|  1|[this, was, a, wa...|[waste, money, wa...|
|Surrender (Bound ...|  2|[surrender, bound...|[surrender, bound...|
|Profanity that ad...|  3|[profanity, that,...|[profanity, adds,...|
|Sad I couldn't co...|  4|[sad, i, couldn, ...|[sad, couldn, con...|
+--------------------+---+--------------------+--------------------+
only showing top 5 rows

+--------------------+---+--------------------+--------------------+------------
--------+
|         review_text| id|               words|            filtered|
filtered_more|
+--------------------+---+--------------------+--------------------+------------
--------+
|Meh Just Ok. 20 r...|  0|[meh, just, ok, 2...|[meh, ok, 20, rec...|[meh, ok,
20, rec...|
|This was a waste ...|  1|[this, was, a, wa...|[waste, money, wa...|[waste,
money, wa...|
|Surrender (Bound ...|  2|[surrender, bound...|[surrender, bound...|[surrender,
bound...|
|Profanity that ad...|  3|[profanity, that,...|[profanity, adds,...|[profanity,
adds,...|
|Sad I couldn't co...|  4|[sad, i, couldn, ...|[sad, couldn, con...|[sad,
couldn, con...|
+--------------------+---+--------------------+--------------------+------------
--------+
only showing top 5 rows
```

```
cv = CountVectorizer(inputCol="filtered_more", outputCol="features", vocabSize = 10000)
cvmodel = cv.fit(tokenized_df2)
featurized_df = cvmodel.transform(tokenized_df2)
vocab = cvmodel.vocabulary
featurized_df.select('filtered_more','features','id').show(5)

countVectors = featurized_df.select('features','id')
countVectors.persist()
print('Records in the DF:', countVectors.count())
```

Output:

```
+--------------------+--------------------+---+
|       filtered_more|            features| id|
+--------------------+--------------------+---+
|[meh, ok, 20, rec...|(10000,[45,69,151...|  0|
|[waste, money, wa...|(10000,[0,2,23,24...|  1|
|[surrender, bound...|(10000,[0,1,2,3,7...|  2|
|[profanity, adds,...|(10000,[0,1,3,5,6...|  3|
|[sad, couldn, con...|(10000,[5,8,22,27...|  4|
+--------------------+--------------------+---+
only showing top 5 rows

Records in the DF: 38894
```

```
countVectors = featurized_df.select('features','id')
countVectors.persist()
print('Records in the DF:', countVectors.count())
```

Output:

```
Records in the DF: 38894
```

```
lda = LDA(k=10, maxIter=5)
model = lda.fit(countVectors)
```

4.3 Identify 5 top topics for 1/2 rating

```
topics = model.describeTopics()
topics_rdd = topics.rdd

topics_words = topics_rdd\
       .map(lambda row: row['termIndices'])\
       .map(lambda idx_list: [vocab[idx] for idx in idx_list])\
       .collect()

for idx, topic in enumerate(topics_words):
    print ("topic: ", idx)
    print ("----------")
    for word in topic:
```

```
        print (word)
    print ("----------")
```

Output:

```
topic:  0
----------
author
story
good
like
people
time
read
way
books
written
----------
topic:  1
----------
read
like
time
author
story
books
really
didn
money
good
----------
topic:  2
----------
read
really
time
like
story
thought
characters
star
writing
series
----------
topic:  3
----------
story
characters
like
really
read
good
reading
character
love
didn
----------
```

```
topic:  4
----------
like
story
really
read
characters
author
free
reading
finish
way
----------
topic:  5
----------
stars
kindle
read
written
like
boring
star
reading
information
bad
----------
topic:  6
----------
read
books
reading
like
star
time
boring
good
better
characters
----------
topic:  7
----------
like
que
story
time
blah
money
little
characters
reading
character
----------
topic:  8
----------
story
good
didn
like
```

```
really
way
read
series
characters
books
----------
topic:  9
----------
star
read
books
like
author
reading
character
written
good
way
----------
```

4.4 Does topic modeling provides good approximation to number of stars given in the review?

We can see there are some positive words in the output even with reviews having star ratings less than 3. Hence in this case topic modelling might not be so effective.

Conclusion:

After performing detailed analysis of "Digital eBook Purchase" versus Books, we can interpret number of reviews for Digital E-book Purchase category are more as compared to Books category and average star rating for books category is high as compared to digital e-book purchase.

After performing analysis to Identify similar products (books) in both categories. Use "product_title" to match products we can interpret that printed book has got more number of higher rating. Hence count of more than 4 star ratings is higher for printed books as compared to digital book star ratings.

After performing LDA to analyze effectiveness of topic modelling in both star rating 4/5and 1/2 cases, we e can see there are some positive words in the output even with reviews having star ratings less than 3. Hence in this case topic modelling might not be so effective.

References:

Amazon reviews dataset: https://registry.opendata.aws/amazon-reviews/

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

Code Reference: https://spark.apache.org/docs/latest/api/python/index.html