brown_analytics

November 28, 2024

0.0.1 DATA 1050 Spark Project

Name: Shiyu Liu

[ ]:
```
from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .appName("Load CSV into Tables") \
    .config("spark.sql.catalogImplementation", "hive") \
    .enableHiveSupport() \
    .getOrCreate()
```

```
your 131072x1 screen size is bogus. expect trouble
24/11/28 14:28:20 WARN Utils: Your hostname, LAPTOP-8R881B48 resolves to a
loopback address: 127.0.1.1; using 10.255.255.254 instead (on interface lo)
24/11/28 14:28:20 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
24/11/28 14:28:20 WARN NativeCodeLoader: Unable to load native-hadoop library
for your platform… using builtin-java classes where applicable

24/11/28 14:28:33 WARN GarbageCollectionMetrics: To enable non-built-in garbage
collector(s) List(G1 Concurrent GC), users should configure it(them) to
spark.eventLog.gcMetrics.youngGenerationGarbageCollectors or
spark.eventLog.gcMetrics.oldGenerationGarbageCollectors
```

[4]:
```
# Load log.csv
log = spark.read.csv('log.csv', header=False, inferSchema=True)
names_list = ["sentiment", "publication_URL", "product_URL", "got_click",␣
 ↪"gender", "age_group"]
log = log.toDF(*names_list)
log.show(5)
log.printSchema()
```

```
+---------+--------------------+--------------------+---------+----------+------
----+
|sentiment|     publication_URL|         product_URL|got_click|    gender|
age_group|
```

```
+---------+--------------------+--------------------+---------+----------+------
----+
| positive|https://www.foxne…|https://lees.com/…|        0|    female|
juvenile|
|  neutral|https://www.mirro…|https://coach.com…|        0|      male|
young|
| negative|https://www.nbcne…|https://covergirl…|        0|
male|middle-age|
| positive|https://www.exami…|https://covergirl…|        0|non-binary|
juvenile|
| negative|  https://www.nj.com|https://dell.com/…|        1|    female|
young|
+---------+--------------------+--------------------+---------+----------+------
----+
only showing top 5 rows

root
 |-- sentiment: string (nullable = true)
 |-- publication_URL: string (nullable = true)
 |-- product_URL: string (nullable = true)
 |-- got_click: integer (nullable = true)
 |-- gender: string (nullable = true)
 |-- age_group: string (nullable = true)
```

[7]:
```
# Load product_categories.csv
product_cat = spark.read.csv('product_categories.csv', header=True,
 ↪inferSchema=True)
product_cat.show(5)
product_cat.printSchema()
```

```
+---------------+--------------------+
|        product|            category|
+---------------+--------------------+
|        blender| small kitchen ap…|
|pressure cooker| small kitchen ap…|
|       computer| consumer electro…|
|         coffee|      packaged food|
|        vitamin|              health|
+---------------+--------------------+
only showing top 5 rows

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

[8]:
```python
# Load products.csv
products = spark.read.csv('products.csv', header=True, inferSchema=True)
products.show(5)
products.printSchema()
```

```
+--------------------+--------------------+---------------+
|             product|         product_URL|   product_type|
+--------------------+--------------------+---------------+
|     Vitamix blender| https://vitamix…|        blender|
|       Lenova laptop| https://lenova.c…|       computer|
|InstantPot pressu…|https://InstantPo…|pressure cooker|
|       NemoK blender|http://nemoK.co/b…|        blender|
|Hamilton Beach bl…|https://HamiltonB…|        blender|
+--------------------+--------------------+---------------+
only showing top 5 rows

root
 |-- product: string (nullable = true)
 |-- product_URL: string (nullable = true)
 |-- product_type: string (nullable = true)
```

[37]:
```python
# For each product, compute all the Publication_URLs containing an ad for that␣
↪product.
from pyspark.sql.functions import countDistinct

product_log = log.join(products, on="product_URL", how="inner")
publication_count = product_log.groupBy("product").agg(
    countDistinct("publication_URL").alias("distinct_count")
)
print(publication_count.count())
publication_count.show(n=publication_count.count(), truncate=False)
```

```
46
+--------------------------+--------------+
|product                   |distinct_count|
+--------------------------+--------------+
|Coach purse               |18            |
|Docker pants              |13            |
|Remington shaver          |11            |
|Maytag washer             |21            |
|Ikea sofa                 |11            |
|Levis Jeans               |17            |
|Samsung TV                |12            |
|covergirl lipstick        |11            |
|Covergirl makeup          |16            |
|Haier refrigerator        |10            |
|Centrum MultiVitamins     |19            |
```

```
|Apple iPad              |17            |
|Soundwave speakers      |14            |
|LG washer               |14            |
|Samsung washer          |19            |
|Maytag refrigerator     |9             |
|Samsung dryer           |13            |
|Ford  sedan             |15            |
|bose speakers           |15            |
|LG TV                   |13            |
|BasilBasel perfume      |15            |
|Apple laptop            |10            |
|Givenchy perfume        |17            |
|Tesla                   |17            |
|Gillette shaver         |11            |
|NemoK blender           |17            |
|LG dryer                |12            |
|Dell computer           |16            |
|Lee jeans               |15            |
|Maybelline lipstick     |15            |
|NordicTrack treadmill   |19            |
|Cougar jeans            |17            |
|Lavazza Coffee          |9             |
|Kaai handbags           |15            |
|Broyhill recliner       |16            |
|Maytag dryer            |14            |
|Dell laptop             |22            |
|Guess perfume           |13            |
|Giorgio perfume         |14            |
|Jaguar perfume          |12            |
|NordicTrack rower       |17            |
|Starbucks Coffee        |19            |
|Apple computer          |14            |
|Sony TV                 |13            |
|InstantPot pressure cooker|14          |
|Clinique moisturizer    |16            |
+------------------------+--------------+
```

[38]:

```
# For each product type, compute all the Publication_URLs containing an ad for␣
↪that product type.
product_type = product_log.groupBy("product_type").agg(
    countDistinct("publication_URL").alias("distinct_count")
)
print(product_type.count())
product_type.show(n=product_type.count(), truncate=False)
```

```
23
+---------------+--------------+
```

```
|product_type   |distinct_count|
+---------------+--------------+
|television     |24            |
|refrigerator   |17            |
|lipstick       |21            |
|face cream     |16            |
|vitamin        |19            |
|tablet         |17            |
|rowing machine |17            |
|pressure cooker|14            |
|shaver         |18            |
|washer         |34            |
|makeup         |16            |
|women's purse  |24            |
|jeans          |31            |
|furniture      |23            |
|dryer          |27            |
|computer       |35            |
|coffee         |23            |
|speakers       |23            |
|car            |25            |
|treadmill      |19            |
|perfume        |36            |
|blender        |17            |
|pants          |13            |
+---------------+--------------+
```

[40]:

```python
# For each product, compute the click rate for it. Click rate is the number of␣
↪times a display of
# an ad was clicked on (by any user) divided by the number of times it was␣
↪displayed (to any user).
# Note the click rate is not specific to each user.
from pyspark.sql.functions import col, sum, count

product_click = product_log.groupBy("product").agg(
    sum(col("got_click")).alias("total_clicks"),
    count("*").alias("total_count")
)
product_click = product_click.withColumn(
    "click_rate", col("total_clicks") / col("total_count")
)
print(product_click.count())
product_click.show(n=product_click.count(), truncate=False)
```

```
46
+--------------------------+------------+-----------+------------------+
|product                   |total_clicks|total_count|click_rate        |
```

Coach purse	89	229	0.388646288209607
Docker pants	106	155	0.6838709677419355
Remington shaver	50	143	0.34965034965034963
Maytag washer	146	288	0.5069444444444444
Ikea sofa	102	178	0.5730337078651685
Levis Jeans	151	226	0.668141592920354
Samsung TV	116	159	0.7295597484276729
covergirl lipstick	112	137	0.8175182481751825
Covergirl makeup	51	202	0.2524752475247525
Haier refrigerator	33	159	0.20754716981132076
Centrum MultiVitamins	151	241	0.6265560165975104
Apple iPad	133	264	0.5037878787878788
Soundwave speakers	113	207	0.5458937198067633
LG washer	106	214	0.4953271028037383
Samsung washer	146	264	0.553030303030303
Samsung dryer	68	156	0.4358974358974359
Maytag refrigerator	46	116	0.39655172413793105
Ford sedan	31	227	0.13656387665198239
bose speakers	103	196	0.5255102040816326
BasilBasel perfume	100	154	0.6493506493506493
LG TV	75	156	0.4807692307692308
Apple laptop	70	124	0.5645161290322581
Givenchy perfume	97	211	0.4597156398104265
Tesla	142	240	0.5916666666666667
Gillette shaver	112	157	0.7133757961783439
NemoK blender	139	244	0.569672131147541
LG dryer	87	138	0.6304347826086957
Dell computer	144	221	0.6515837104072398
Lee jeans	99	212	0.4669811320754717
Maybelline lipstick	115	205	0.5609756097560976
NordicTrack treadmill	119	243	0.4897119341563786
Cougar jeans	71	273	0.2600732600732601
Lavazza Coffee	66	117	0.5641025641025641
Kaai handbags	132	200	0.66
Broyhill recliner	110	204	0.5392156862745098
Maytag dryer	70	203	0.3448275862068966
Dell laptop	220	698	0.3151862464183381
Guess perfume	70	172	0.4069767441860465
Giorgio perfume	171	214	0.7990654205607477
Jaguar perfume	62	131	0.4732824427480916
NordicTrack rower	42	188	0.22340425531914893
Starbucks Coffee	85	308	0.275974025974026
Apple computer	160	202	0.7920792079207921
InstantPot pressure cooker	96	192	0.5
Sony TV	66	168	0.39285714285714285
Clinique moisturizer	174	216	0.8055555555555556

[39]:

```python
# For each product, compute the click rate for each sentiment type
product_sent_click = product_log.groupBy("product", "sentiment").agg(
    sum(col("got_click")).alias("total_clicks"),
    count("*").alias("total_count")
)
product_sent_click = product_sent_click.withColumn(
    "click_rate", col("total_clicks") / col("total_count")
)
product_sent_click = product_sent_click.orderBy("product", "sentiment")
print(product_sent_click.count())
product_sent_click.show(n=product_sent_click.count(), truncate=False)
```

```
138
+---------------------------+---------+------------+-----------+-----------------
---+
|product                    |sentiment|total_clicks|total_count|click_rate
|
+---------------------------+---------+------------+-----------+-----------------
---+
|Apple computer             |negative |48          |69
|0.6956521739130435  |
|Apple computer             |neutral  |60          |65
|0.9230769230769231  |
|Apple computer             |positive |52          |68
|0.7647058823529411  |
|Apple iPad                 |negative |36          |92         |0.391304347826087
|
|Apple iPad                 |neutral  |54          |86         |0.627906976744186
|
|Apple iPad                 |positive |43          |86         |0.5
|
|Apple laptop               |negative |6           |39
|0.15384615384615385 |
|Apple laptop               |neutral  |30          |41
|0.7317073170731707  |
|Apple laptop               |positive |34          |44
|0.7727272727272727  |
|BasilBasel perfume         |negative |27          |35
|0.7714285714285715  |
|BasilBasel perfume         |neutral  |51          |59         |0.864406779661017
|
|BasilBasel perfume         |positive |22          |60
|0.36666666666666664 |
|Broyhill recliner          |negative |57          |70
|0.8142857142857143  |
|Broyhill recliner          |neutral  |36          |64         |0.5625
```

```
|
|Broyhill recliner          |positive |17           |70
|0.24285714285714285 |
|Centrum MultiVitamins     |negative |71           |85
|0.8352941176470589  |
|Centrum MultiVitamins     |neutral  |66           |81
|0.8148148148148148  |
|Centrum MultiVitamins     |positive |14           |75
|0.18666666666666668 |
|Clinique moisturizer      |negative |65           |72
|0.9027777777777778  |
|Clinique moisturizer      |neutral  |67           |72
|0.9305555555555556  |
|Clinique moisturizer      |positive |42           |72
|0.5833333333333334  |
|Coach purse               |negative |25           |79
|0.31645569620253167 |
|Coach purse               |neutral  |34           |75
|0.4533333333333333  |
|Coach purse               |positive |30           |75           |0.4
|
|Cougar jeans              |negative |8            |96
|0.08333333333333333 |
|Cougar jeans              |neutral  |24           |77
|0.3116883116883117  |
|Cougar jeans              |positive |39           |100          |0.39
|
|Covergirl makeup          |negative |8            |71
|0.11267605633802817 |
|Covergirl makeup          |neutral  |30           |78
|0.38461538461538464 |
|Covergirl makeup          |positive |13           |53
|0.24528301886792453 |
|Dell computer             |negative |59           |71
|0.8309859154929577  |
|Dell computer             |neutral  |26           |68
|0.38235294117647056 |
|Dell computer             |positive |59           |82
|0.7195121951219512  |
|Dell laptop               |negative |34           |234
|0.1452991452991453  |
|Dell laptop               |neutral  |90           |252
|0.35714285714285715 |
|Dell laptop               |positive |96           |212
|0.4528301886792453  |
|Docker pants              |negative |56           |67           |0.835820895522388
|
|Docker pants              |neutral  |35           |43           |0.813953488372093
```

```
|
|Docker pants              |positive |15          |45
|0.3333333333333333  |
|Ford  sedan               |negative |1           |76
|0.013157894736842105|
|Ford  sedan               |neutral  |9           |72          |0.125
|
|Ford  sedan               |positive |21          |79
|0.26582278481012656 |
|Gillette shaver           |negative |57          |63
|0.9047619047619048  |
|Gillette shaver           |neutral  |49          |51
|0.9607843137254902  |
|Gillette shaver           |positive |6           |43
|0.13953488372093023 |
|Giorgio perfume           |negative |53          |83
|0.6385542168674698  |
|Giorgio perfume           |neutral  |64          |66
|0.9696969696969697  |
|Giorgio perfume           |positive |54          |65
|0.8307692307692308  |
|Givenchy perfume          |negative |50          |66
|0.7575757575757576  |
|Givenchy perfume          |neutral  |23          |76
|0.3026315789473684  |
|Givenchy perfume          |positive |24          |69
|0.34782608695652173 |
|Guess perfume             |negative |10          |58
|0.1724137931034483  |
|Guess perfume             |neutral  |19          |62
|0.3064516129032258  |
|Guess perfume             |positive |41          |52
|0.7884615384615384  |
|Haier refrigerator        |negative |9           |52
|0.17307692307692307 |
|Haier refrigerator        |neutral  |9           |54
|0.16666666666666666 |
|Haier refrigerator        |positive |15          |53
|0.2830188679245283  |
|Ikea sofa                 |negative |42          |50          |0.84
|
|Ikea sofa                 |neutral  |41          |69
|0.5942028985507246  |
|Ikea sofa                 |positive |19          |59
|0.3220338983050847  |
|InstantPot pressure cooker|negative |0           |70          |0.0
|
|InstantPot pressure cooker|neutral  |51          |59          |0.864406779661017
```

```
|
|InstantPot pressure cooker|positive |45          |63
|0.7142857142857143  |
|Jaguar perfume            |negative |20          |46
|0.43478260869565216 |
|Jaguar perfume            |neutral  |28          |40          |0.7
|
|Jaguar perfume            |positive |14          |45
|0.3111111111111111  |
|Kaai handbags             |negative |26          |53
|0.49056603773584906 |
|Kaai handbags             |neutral  |66          |69
|0.9565217391304348  |
|Kaai handbags             |positive |40          |78
|0.5128205128205128  |
|LG TV                     |negative |20          |67
|0.29850746268656714 |
|LG TV                     |neutral  |26          |44
|0.5909090909090909  |
|LG TV                     |positive |29          |45
|0.6444444444444445  |
|LG dryer                  |negative |32          |47
|0.6808510638297872  |
|LG dryer                  |neutral  |38          |48
|0.7916666666666666  |
|LG dryer                  |positive |17          |43
|0.3953488372093023  |
|LG washer                 |negative |12          |78
|0.15384615384615385 |
|LG washer                 |neutral  |31          |60
|0.5166666666666667  |
|LG washer                 |positive |63          |76
|0.8289473684210527  |
|Lavazza Coffee            |negative |37          |44
|0.8409090909090909  |
|Lavazza Coffee            |neutral  |17          |38
|0.4473684210526316  |
|Lavazza Coffee            |positive |12          |35
|0.34285714285714286 |
|Lee jeans                 |negative |25          |73
|0.3424657534246575  |
|Lee jeans                 |neutral  |49          |63
|0.7777777777777778  |
|Lee jeans                 |positive |25          |76
|0.32894736842105265 |
|Levis Jeans               |negative |34          |84
|0.40476190476190477 |
|Levis Jeans               |neutral  |52          |65          |0.8
```

```
|
|Levis Jeans              |positive |65          |77
|0.8441558441558441  |
|Maybelline lipstick      |negative |54          |61
|0.8852459016393442  |
|Maybelline lipstick      |neutral  |17          |81
|0.20987654320987653 |
|Maybelline lipstick      |positive |44          |63
|0.6984126984126984  |
|Maytag dryer             |negative |8           |68
|0.11764705882352941 |
|Maytag dryer             |neutral  |43          |61
|0.7049180327868853  |
|Maytag dryer             |positive |19          |74
|0.25675675675675674 |
|Maytag refrigerator      |negative |2           |38
|0.05263157894736842 |
|Maytag refrigerator      |neutral  |26          |36
|0.7222222222222222  |
|Maytag refrigerator      |positive |18          |42
|0.42857142857142855 |
|Maytag washer            |negative |83          |90
|0.9222222222222223  |
|Maytag washer            |neutral  |34          |94
|0.3617021276595745  |
|Maytag washer            |positive |29          |104
|0.27884615384615385 |
|NemoK blender            |negative |90          |92
|0.9782608695652174  |
|NemoK blender            |neutral  |29          |89
|0.3258426966292135  |
|NemoK blender            |positive |20          |63
|0.31746031746031744 |
|NordicTrack rower        |negative |13          |65          |0.2
|
|NordicTrack rower        |neutral  |7           |49
|0.14285714285714285 |
|NordicTrack rower        |positive |22          |74
|0.2972972972972973  |
|NordicTrack treadmill    |negative |25          |79
|0.31645569620253167 |
|NordicTrack treadmill    |neutral  |52          |89
|0.5842696629213483  |
|NordicTrack treadmill    |positive |42          |75          |0.56
|
|Remington shaver         |negative |9           |62
|0.14516129032258066 |
|Remington shaver         |neutral  |9           |38
```

```
|0.23684210526315788 |
|Remington shaver          |positive |32          |43
|0.7441860465116279  |
|Samsung TV                |negative |30          |45
|0.6666666666666666  |
|Samsung TV                |neutral  |41          |50          |0.82
|
|Samsung TV                |positive |45          |64          |0.703125
|
|Samsung dryer             |negative |16          |58
|0.27586206896551724 |
|Samsung dryer             |neutral  |38          |53
|0.7169811320754716  |
|Samsung dryer             |positive |14          |45
|0.3111111111111111  |
|Samsung washer            |negative |0           |84          |0.0
|
|Samsung washer            |neutral  |58          |82
|0.7073170731707317  |
|Samsung washer            |positive |88          |98
|0.8979591836734694  |
|Sony TV                   |negative |3           |54
|0.05555555555555555 |
|Sony TV                   |neutral  |38          |56
|0.6785714285714286  |
|Sony TV                   |positive |25          |58
|0.43103448275862066 |
|Soundwave speakers        |negative |8           |67
|0.11940298507462686 |
|Soundwave speakers        |neutral  |64          |67
|0.9552238805970149  |
|Soundwave speakers        |positive |41          |73
|0.5616438356164384  |
|Starbucks Coffee          |negative |30          |100         |0.3
|
|Starbucks Coffee          |neutral  |33          |102
|0.3235294117647059  |
|Starbucks Coffee          |positive |22          |106
|0.20754716981132076 |
|Tesla                     |negative |57          |72
|0.7916666666666666  |
|Tesla                     |neutral  |80          |81
|0.9876543209876543  |
|Tesla                     |positive |5           |87
|0.05747126436781609 |
|bose speakers             |negative |37          |62
|0.5967741935483871  |
|bose speakers             |neutral  |32          |69          |0.463768115942029
```

```
|
|bose speakers            |positive |34          |65
|0.5230769230769231   |
|covergirl lipstick       |negative |18          |43
|0.4186046511627907   |
|covergirl lipstick       |neutral  |48          |48          |1.0
|
|covergirl lipstick       |positive |46          |46          |1.0
|
+-------------------------+---------+------------+-----------+----------------
---+
```

[36]:
```python
# For each product type, compute the click rate for it.
type_click = product_log.groupBy("product_type").agg(
    sum(col("got_click")).alias("total_clicks"),
    count("*").alias("total_count")
)
type_click = type_click.withColumn(
    "click_rate", col("total_clicks") / col("total_count")
)
print(type_click.count())
type_click.show(n=type_click.count(), truncate=False)
```

```
23
+---------------+------------+-----------+-------------------+
|product_type   |total_clicks|total_count|click_rate         |
+---------------+------------+-----------+-------------------+
|television     |257         |483        |0.5320910973084886 |
|refrigerator   |79          |275        |0.2872727272727273 |
|lipstick       |227         |342        |0.6637426900584795 |
|face cream     |174         |216        |0.8055555555555556 |
|vitamin        |151         |241        |0.6265560165975104 |
|tablet         |133         |264        |0.5037878787878788 |
|rowing machine |42          |188        |0.22340425531914893|
|pressure cooker|96          |192        |0.5                |
|shaver         |162         |300        |0.54               |
|washer         |398         |766        |0.5195822454308094 |
|makeup         |51          |202        |0.2524752475247525 |
|women's purse  |221         |429        |0.5151515151515151 |
|jeans          |321         |711        |0.45147679324894513|
|furniture      |212         |382        |0.5549738219895288 |
|dryer          |225         |497        |0.45271629778672035|
|computer       |594         |1245       |0.4771084337349398 |
|coffee         |151         |425        |0.3552941176470588 |
|speakers       |216         |403        |0.5359801488833746 |
|car            |173         |467        |0.37044967880085655|
|treadmill      |119         |243        |0.4897119341563786 |
```

```
|perfume        |500        |882        |0.5668934240362812 |
|blender        |139        |244        |0.569672131147541  |
|pants          |106        |155        |0.6838709677419355 |
+---------------+-----------+-----------+-------------------+
```

[35]:

```python
# For each product type, compute the click rate for each sentiment type
type_sent_click = product_log.groupBy("product_type", "sentiment").agg(
    sum(col("got_click")).alias("total_clicks"),
    count("*").alias("total_count")
)
type_sent_click = type_sent_click.withColumn(
    "click_rate", col("total_clicks") / col("total_count")
)
type_sent_click = type_sent_click.orderBy("product_type", "sentiment")
print(type_sent_click.count())
type_sent_click.show(n=type_sent_click.count(), truncate=False)
```

```
69
+---------------+---------+------------+-----------+-------------------+
|product_type   |sentiment|total_clicks|total_count|click_rate         |
+---------------+---------+------------+-----------+-------------------+
|blender        |negative |90          |92         |0.9782608695652174 |
|blender        |neutral  |29          |89         |0.3258426966292135 |
|blender        |positive |20          |63         |0.31746031746031744|
|car            |negative |58          |148        |0.3918918918918919 |
|car            |neutral  |89          |153        |0.5816993464052288 |
|car            |positive |26          |166        |0.1566265060240964 |
|coffee         |negative |67          |144        |0.4652777777777778 |
|coffee         |neutral  |50          |140        |0.35714285714285715|
|coffee         |positive |34          |141        |0.24113475177304963|
|computer       |negative |147         |413        |0.3559322033898305 |
|computer       |neutral  |206         |426        |0.4835680751173709 |
|computer       |positive |241         |406        |0.5935960591133005 |
|dryer          |negative |56          |173        |0.3236994219653179 |
|dryer          |neutral  |119         |162        |0.7345679012345679 |
|dryer          |positive |50          |162        |0.30864197530864196|
|face cream     |negative |65          |72         |0.9027777777777778 |
|face cream     |neutral  |67          |72         |0.9305555555555556 |
|face cream     |positive |42          |72         |0.5833333333333334 |
|furniture      |negative |99          |120        |0.825              |
|furniture      |neutral  |77          |133        |0.5789473684210527 |
|furniture      |positive |36          |129        |0.27906976744186046|
|jeans          |negative |67          |253        |0.2648221343873518 |
|jeans          |neutral  |125         |205        |0.6097560975609756 |
|jeans          |positive |129         |253        |0.5098814229249012 |
|lipstick       |negative |72          |104        |0.6923076923076923 |
|lipstick       |neutral  |65          |129        |0.5038759689922481 |
```

lipstick	positive	90	109	0.8256880733944955
makeup	negative	8	71	0.11267605633802817
makeup	neutral	30	78	0.38461538461538464
makeup	positive	13	53	0.24528301886792453
pants	negative	56	67	0.835820895522388
pants	neutral	35	43	0.813953488372093
pants	positive	15	45	0.3333333333333333
perfume	negative	160	288	0.5555555555555556
perfume	neutral	185	303	0.6105610561056105
perfume	positive	155	291	0.5326460481099656
pressure cooker	negative	0	70	0.0
pressure cooker	neutral	51	59	0.864406779661017
pressure cooker	positive	45	63	0.7142857142857143
refrigerator	negative	11	90	0.12222222222222222
refrigerator	neutral	35	90	0.3888888888888889
refrigerator	positive	33	95	0.3473684210526316
rowing machine	negative	13	65	0.2
rowing machine	neutral	7	49	0.14285714285714285
rowing machine	positive	22	74	0.2972972972972973
shaver	negative	66	125	0.528
shaver	neutral	58	89	0.651685393258427
shaver	positive	38	86	0.4418604651162791
speakers	negative	45	129	0.3488372093023256
speakers	neutral	96	136	0.7058823529411765
speakers	positive	75	138	0.5434782608695652
tablet	negative	36	92	0.391304347826087
tablet	neutral	54	86	0.627906976744186
tablet	positive	43	86	0.5
television	negative	53	166	0.3192771084337349
television	neutral	105	150	0.7
television	positive	99	167	0.592814371257485
treadmill	negative	25	79	0.31645569620253167
treadmill	neutral	52	89	0.5842696629213483
treadmill	positive	42	75	0.56
vitamin	negative	71	85	0.8352941176470589
vitamin	neutral	66	81	0.8148148148148148
vitamin	positive	14	75	0.18666666666666668
washer	negative	95	252	0.376984126984127
washer	neutral	123	236	0.5211864406779662
washer	positive	180	278	0.6474820143884892
women's purse	negative	51	132	0.38636363636363635
women's purse	neutral	100	144	0.6944444444444444
women's purse	positive	70	153	0.45751633986928103

[31]:
```python
# For each category, compute the click rate for it.
product_log_cat = product_log.join(product_cat, product_log.product_type ==␣
 ↪product_cat.product, how="inner")
cat_click = product_log_cat.groupBy("category").agg(
    sum(col("got_click")).alias("total_clicks"),
    count("*").alias("total_count")
)
cat_click = cat_click.withColumn(
    "click_rate", col("total_clicks") / col("total_count")
)
print(cat_click.count())
cat_click.show()
```

```
12
+--------------------+------------+-----------+-------------------+
|            category|total_clicks|total_count|         click_rate|
+--------------------+------------+-----------+-------------------+
|     beauty products|         952|       1642| 0.5797807551766139|
|large kitchen app…|         702|       1538| 0.4564369310793238|
|             apparel|         427|        866| 0.4930715935334873|
| small kitchen ap…|         235|        436| 0.5389908256880734|
|   fitness equipment|         161|        431|0.37354988399071926|
|              health|         151|        241| 0.6265560165975104|
|         accessories|         221|        429| 0.5151515151515151|
|consumer electronics|         768|       1450| 0.5296551724137931|
|        packaged food|         151|        425| 0.3552941176470588|
|   household durables|         212|        382| 0.5549738219895288|
|      transportation|         173|        467|0.37044967880085655|
| consumer electro…|         594|       1245| 0.4771084337349398|
+--------------------+------------+-----------+-------------------+
```

[41]:
```python
# Similar to above, for each category compute the click rate for each sentiment␣
 ↪type
cat_sent_click = product_log_cat.groupBy("category","sentiment").agg(
    sum(col("got_click")).alias("total_clicks"),
    count("*").alias("total_count")
)
cat_sent_click = cat_sent_click.withColumn(
    "click_rate", col("total_clicks") / col("total_count")
)
cat_sent_click = cat_sent_click.orderBy("category", "sentiment")
print(cat_sent_click.count())
cat_sent_click.show(n=cat_sent_click.count(), truncate=False)
```

```
36
+-------------------------+---------+------------+-----------+------------------
-+
```

```
|category                 |sentiment|total_clicks|total_count|click_rate
|
+-------------------------+---------+------------+-----------+------------------
-+
| consumer electronics    |negative |147         |413        |0.3559322033898305
|
| consumer electronics    |neutral  |206         |426        |0.4835680751173709
|
| consumer electronics    |positive |241         |406        |0.5935960591133005
|
| packaged food           |negative |67          |144        |0.4652777777777778
|
| packaged food           |neutral  |50          |140
|0.35714285714285715|
| packaged food           |positive |34          |141
|0.24113475177304963|
| small kitchen appliances|negative |90          |162        |0.5555555555555556
|
| small kitchen appliances|neutral  |80          |148        |0.5405405405405406
|
| small kitchen appliances|positive |65          |126        |0.5158730158730159
|
|accessories              |negative |51          |132
|0.38636363636363635|
|accessories              |neutral  |100         |144        |0.6944444444444444
|
|accessories              |positive |70          |153
|0.45751633986928103|
|apparel                  |negative |123         |320        |0.384375
|
|apparel                  |neutral  |160         |248        |0.6451612903225806
|
|apparel                  |positive |144         |298
|0.48322147651006714|
|beauty products          |negative |305         |535        |0.5700934579439252
|
|beauty products          |neutral  |347         |582        |0.5962199312714777
|
|beauty products          |positive |300         |525        |0.5714285714285714
|
|consumer electronics     |negative |200         |512        |0.390625
|
|consumer electronics     |neutral  |313         |461        |0.6789587852494577
|
|consumer electronics     |positive |255         |477        |0.5345911949685535
|
|fitness equipment        |negative |38          |144        |0.2638888888888889
|
```

```
|fitness equipment        |neutral  |59          |138         |0.427536231884058
|
|fitness equipment        |positive |64          |149
|0.42953020134228187|
|health                   |negative |71          |85          |0.8352941176470589
|
|health                   |neutral  |66          |81          |0.8148148148148148
|
|health                   |positive |14          |75
|0.18666666666666668|
|household durables       |negative |99          |120         |0.825
|
|household durables       |neutral  |77          |133         |0.5789473684210527
|
|household durables       |positive |36          |129
|0.27906976744186046|
|large kitchen appliances |negative |162         |515         |0.3145631067961165
|
|large kitchen appliances |neutral  |277         |488         |0.5676229508196722
|
|large kitchen appliances |positive |263         |535         |0.491588785046729
|
|transportation           |negative |58          |148         |0.3918918918918919
|
|transportation           |neutral  |89          |153         |0.5816993464052288
|
|transportation           |positive |26          |166         |0.1566265060240964
|
+-------------------------+---------+------------+-----------+------------------
-+
```

[33]:

```
# Choose a product randomly; determine if there are any 'significant'␣
↪differences
# in the click rate between positive and negative sentiment type of the ad␣
↪context
# for that product type given the gender of the viewer.
starbucks = product_sent_click.filter(col("product") == "Starbucks Coffee")
starbucks.show()
```

```
+----------------+---------+------------+-----------+-------------------+
|         product|sentiment|total_clicks|total_count|         click_rate|
+----------------+---------+------------+-----------+-------------------+
|Starbucks Coffee| negative|          30|        100|                0.3|
|Starbucks Coffee|  neutral|          33|        102| 0.3235294117647059|
|Starbucks Coffee| positive|          22|        106|0.20754716981132076|
+----------------+---------+------------+-----------+-------------------+
```

Assuming a binomial distribution for click rate, apply z-test: - **Null Hypothesis:** There is no significant difference between the click rates for positive and negative sentiment types:

$$H_0 : p_1 = p_2$$

- **Alternative Hypothesis:** There is a significant difference between the click rates for positive and negative sentiment types:

$$H_1 : p_1 \neq p_2$$

[ ]:
```python
from math import sqrt
from scipy.stats import norm

positive = starbucks.filter(col("sentiment") == "positive").collect()[0]
negative = starbucks.filter(col("sentiment") == "negative").collect()[0]

# Positive sentiment stats
p1 = positive["click_rate"]
n1 = positive["total_count"]
x1 = positive["total_clicks"]

# Negative sentiment stats
p2 = negative["click_rate"]
n2 = negative["total_count"]
x2 = negative["total_clicks"]

# Pooled proportion
p_hat = (x1 + x2) / (n1 + n2)

# Standard error
se = sqrt(p_hat * (1 - p_hat) * (1/n1 + 1/n2))

# Z-score
z = (p1 - p2) / se

# Two-tailed p-value
p_value = 2 * (1 - norm.cdf(abs(z)))

print(f"Z-score: {z}")
print(f"P-value: {p_value}")
```

```
Z-score: -1.5266684163098292
P-value: 0.12684348491358133
```

Since $p = 0.1268 > 0.05$, we fail to reject the null hypothesis. This means that the observed difference in click rates between positive and negative sentiments for Starbucks Coffee is not statistically significant at the 5% level.

Extra Credit - Compute the click rate for each age group.

[57]:
```
age_click = log.groupBy()
age_click = log.groupBy("age_group").agg(
    sum(col("got_click")).alias("total_clicks"),
    count("*").alias("total_count")
)
age_click = age_click.withColumn(
    "click_rate", col("total_clicks") / col("total_count")
)
age_click = age_click.orderBy("age_group")
age_click.show(n=age_click.count(), truncate=False)
```

```
+----------+------------+-----------+-------------------+
|age_group |total_clicks|total_count|click_rate         |
+----------+------------+-----------+-------------------+
|juvenile  |1181        |2485       |0.47525150905432595|
|middle-age|1180        |2497       |0.4725670804965959 |
|senior    |1338        |2490       |0.5373493975903615 |
|young     |1354        |2528       |0.5356012658227848 |
+----------+------------+-----------+-------------------+
```

We observe that senior and young people generally have higher click rates than the other two age groups.

- Compute the click rate for each gender.

[58]:
```
gender_click = log.groupBy()
gender_click = log.groupBy("gender").agg(
    sum(col("got_click")).alias("total_clicks"),
    count("*").alias("total_count")
)
gender_click = gender_click.withColumn(
    "click_rate", col("total_clicks") / col("total_count")
)
gender_click = gender_click.orderBy("gender")
gender_click.show(n=gender_click.count(), truncate=False)
```

```
+----------+------------+-----------+------------------+
|gender    |total_clicks|total_count|click_rate        |
+----------+------------+-----------+------------------+
|female    |2564        |5050       |0.5077227722772277|
|male      |2477        |4929       |0.5025360113613309|
|non-binary|12          |21         |0.5714285714285714|
+----------+------------+-----------+------------------+
```

We observe that non-binary individuals have higher click rates compared to the other two genders. However, this difference could be influenced by the smaller sample size for non-binary individuals, which may skew the results.

- Compute the click rate for each sentiment.

[59]:
```python
sent_click = log.groupBy()
sent_click = log.groupBy("sentiment").agg(
    sum(col("got_click")).alias("total_clicks"),
    count("*").alias("total_count")
)
sent_click = sent_click.withColumn(
    "click_rate", col("total_clicks") / col("total_count")
)
sent_click = sent_click.orderBy("sentiment")
sent_click.show(n=sent_click.count(), truncate=False)
```

```
+---------+------------+-----------+-------------------+
|sentiment|total_clicks|total_count|click_rate         |
+---------+------------+-----------+-------------------+
|negative |1566        |3392       |0.46167452830188677|
|neutral  |1922        |3279       |0.5861543153400427 |
|positive |1565        |3329       |0.4701111444878342 |
+---------+------------+-----------+-------------------+
```

We can see that people with neutral sentiment have the highest click rate among all sentiments.

- For each age group, what are the types of product with the top 5 highest ads click rate?

[ ]:
```python
from pyspark.sql.window import Window
from pyspark.sql.functions import rank, col, desc

type_age_click = product_log.groupBy("product_type", "age_group").agg(
    sum(col("got_click")).alias("total_clicks"),
    count("*").alias("total_count")
)
type_age_click = type_age_click.withColumn(
    "click_rate", col("total_clicks") / col("total_count")
)
window_spec = Window.partitionBy("age_group").orderBy(desc("click_rate"))
type_age_click_ranked = type_age_click.withColumn("rank", rank().
 ↪over(window_spec))
top_5_by_age_group = type_age_click_ranked.filter(col("rank") <= 5)
top_5_by_age_group.show()
```

```
+---------------+----------+------------+-----------+------------------+----+
|   product_type| age_group|total_clicks|total_count|        click_rate|rank|
+---------------+----------+------------+-----------+------------------+----+
|     face cream|  juvenile|          43|         54|0.7962962962962963|   1|
|        vitamin|  juvenile|          36|         58|0.6206896551724138|   2|
|  women's purse|  juvenile|          54|         89|0.6067415730337079|   3|
|pressure cooker|  juvenile|          27|         45|               0.6|   4|
|       lipstick|  juvenile|          51|         89|0.5730337078651685|   5|
```

```
|     face cream|middle-age|          44|         53|0.8301886792452831|   1|
|          pants|middle-age|          36|         47|0.7659574468085106|   2|
|        blender|middle-age|          35|         57|0.6140350877192983|   3|
|       lipstick|middle-age|          49|         82|0.5975609756097561|   4|
|        vitamin|middle-age|          31|         54|0.5740740740740741|   5|
|       lipstick|    senior|          63|         79|0.7974683544303798|   1|
|     face cream|    senior|          37|         48|0.7708333333333334|   2|
|          pants|    senior|          28|         41|0.6829268292682927|   3|
|        vitamin|    senior|          39|         62|0.6290322580645161|   4|
|        perfume|    senior|         144|        229|  0.62882096069869|   5|
|     face cream|     young|          50|         61| 0.819672131147541|   1|
|       lipstick|     young|          64|         92|0.6956521739130435|   2|
|          pants|     young|          25|         37|0.6756756756756757|   3|
|        vitamin|     young|          45|         67|0.6716417910447762|   4|
|         washer|     young|         110|        181|0.6077348066298343|   5|
+---------------+----------+------------+-----------+------------------+----+
```

We can see that the ads for face cream have the highest click rate across all age groups, except for senior. Some other types of product with high click rates are pants, vitamin and lipstick.

- Find the top 5 brands with the highest click rates and the top 5 brands with the lowest click rates for each gender.

[60]:
```python
from pyspark.sql.functions import split

product_log = product_log.withColumn(
    "brand",
    split(col("product"), " ")[0]
)

gender_brand_click = product_log.groupBy("gender", "brand").agg(
    sum(col("got_click")).alias("total_clicks"),
    count("*").alias("total_count")
)
gender_brand_click = gender_brand_click.withColumn(
    "click_rate", col("total_clicks") / col("total_count")
)
window_spec = Window.partitionBy("gender").orderBy(desc("click_rate"))
gender_brand_click_ranked = gender_brand_click.withColumn("rank", rank().
 ↪over(window_spec))
top_5_by_gender = gender_brand_click_ranked.filter(col("rank") <= 5)
top_5_by_gender.show()
```

```
+----------+----------+------------+-----------+------------------+----+
|    gender|     brand|total_clicks|total_count|        click_rate|rank|
+----------+----------+------------+-----------+------------------+----+
|    female| covergirl|          53|         65|0.8153846153846154|   1|
|    female|   Giorgio|          85|        105|0.8095238095238095|   2|
```

```
|    female|  Clinique|          93|        119|0.7815126050420168|   3|
|    female|  Gillette|          65|         84|0.7738095238095238|   4|
|    female|    Docker|          51|         72|0.7083333333333334|   5|
|      male|  Clinique|          80|         96|0.8333333333333334|   1|
|      male| covergirl|          59|         72|0.8194444444444444|   2|
|      male|   Giorgio|          86|        109|0.7889908256880734|   3|
|      male|BasilBasel|          53|         77|0.6883116883116883|   4|
|      male|      Kaai|          73|        108|0.6759259259259259|   5|
|non-binary|       Lee|           1|          1|               1.0|   1|
|non-binary| Starbucks|           1|          1|               1.0|   1|
|non-binary|   Samsung|           1|          1|               1.0|   1|
|non-binary|     Apple|           1|          1|               1.0|   1|
|non-binary|  Clinique|           1|          1|               1.0|   1|
|non-binary|      Kaai|           1|          1|               1.0|   1|
|non-binary|    Docker|           1|          1|               1.0|   1|
|non-binary|      bose|           1|          1|               1.0|   1|
+----------+----------+------------+-----------+------------------+----+
```

[61]:

```python
window_spec = Window.partitionBy("gender").orderBy("click_rate")
gender_brand_click_ranked_asc = gender_brand_click.withColumn("rank", rank().
 ↪over(window_spec))
bottom_5_by_gender = gender_brand_click_ranked_asc.filter(col("rank") <= 5)
bottom_5_by_gender.show()
```

```
+----------+-----------+------------+-----------+-------------------+----+
|    gender|      brand|total_clicks|total_count|         click_rate|rank|
+----------+-----------+------------+-----------+-------------------+----+
|    female|       Ford|          14|        115|0.12173913043478261|   1|
|    female|  Covergirl|          21|         98|0.21428571428571427|   2|
|    female|     Cougar|          37|        146| 0.2534246575342466|   3|
|    female|  Remington|          14|         55| 0.2545454545454545|   4|
|    female|      Haier|          22|         83|0.26506024096385544|   5|
|      male|      Haier|          11|         76|0.14473684210526316|   1|
|      male|       Ford|          17|        112|0.15178571428571427|   2|
|      male|  Starbucks|          38|        162| 0.2345679012345679|   3|
|      male|     Cougar|          34|        127| 0.2677165354330709|   4|
|      male|       Sony|          23|         79| 0.2911392405063291|   5|
|non-binary|   Broyhill|           0|          1|                0.0|   1|
|non-binary|  Remington|           0|          1|                0.0|   1|
|non-binary|NordicTrack|           0|          1|                0.0|   1|
|non-binary|   Gillette|           0|          1|                0.0|   1|
|non-binary|  Covergirl|           0|          1|                0.0|   1|
|non-binary|      Guess|           0|          1|                0.0|   1|
|non-binary|     Jaguar|           0|          1|                0.0|   1|
|non-binary|      Coach|           0|          1|                0.0|   1|
+----------+-----------+------------+-----------+-------------------+----+
```