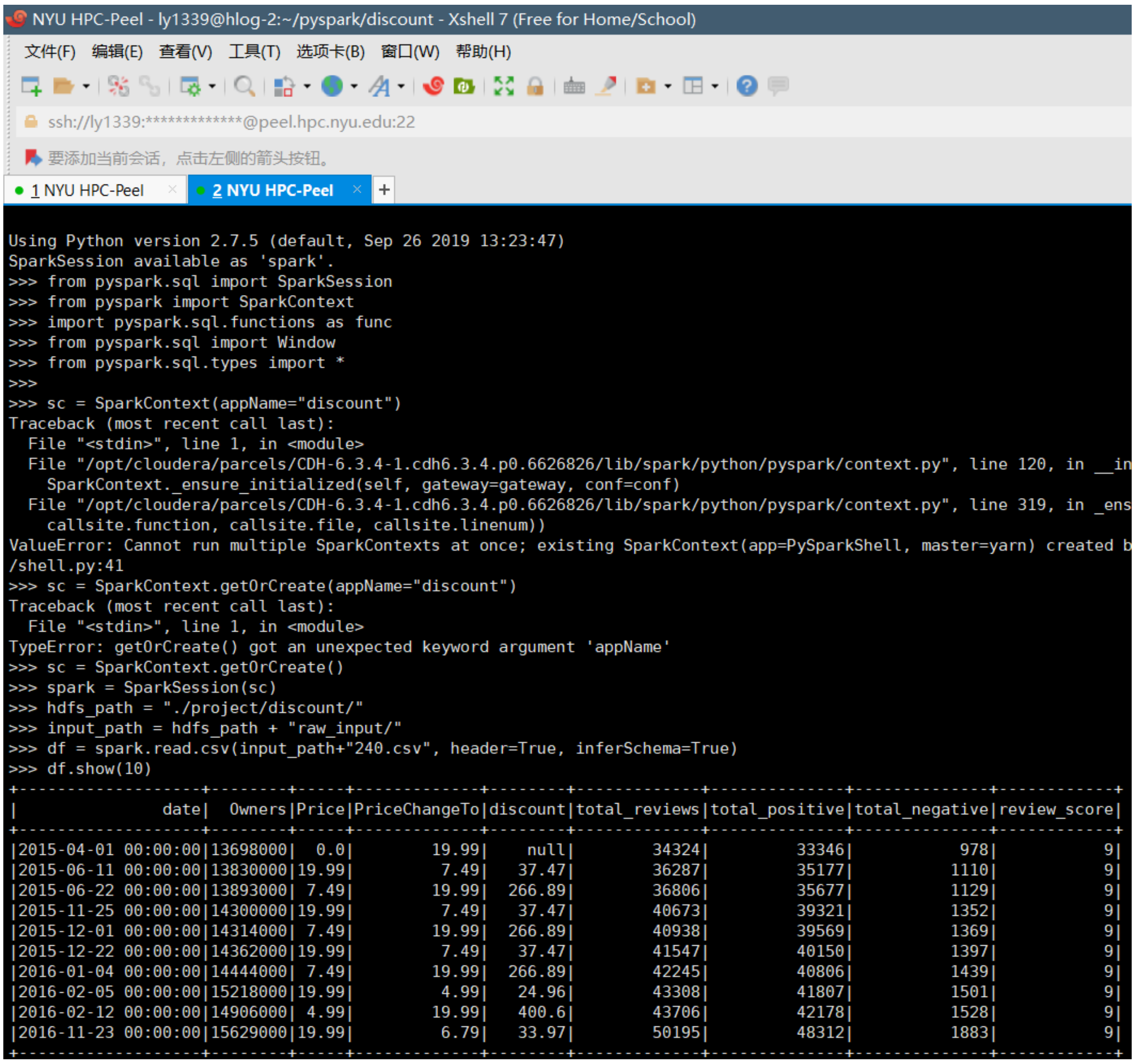
```python
from pyspark.sql import SparkSession
from pyspark import SparkContext
import pyspark.sql.functions as func
from pyspark.sql import Window
from pyspark.sql.types import *

sc = SparkContext.getOrCreate()
spark = SparkSession(sc)
```

```python
# My Net Id: ly1339
hdfs_path = "./project/discount/"
input_path = hdfs_path + "raw_input/"
```

1. Choose a single input as test, Full ETL code is in ELT folder.

```python
df = spark.read.csv(input_path+"240.csv", header=True, inferSchema=True)
df.show(10)
```


NYU HPC-Peel - ly1339@hlog-2:~/pyspark/discount - Xshell 7 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://ly1339:*************@peel.hpc.nyu.edu:22
要添加当前会话，点击左侧的箭头按钮。
1 NYU HPC-Peel
2 NYU HPC-Peel

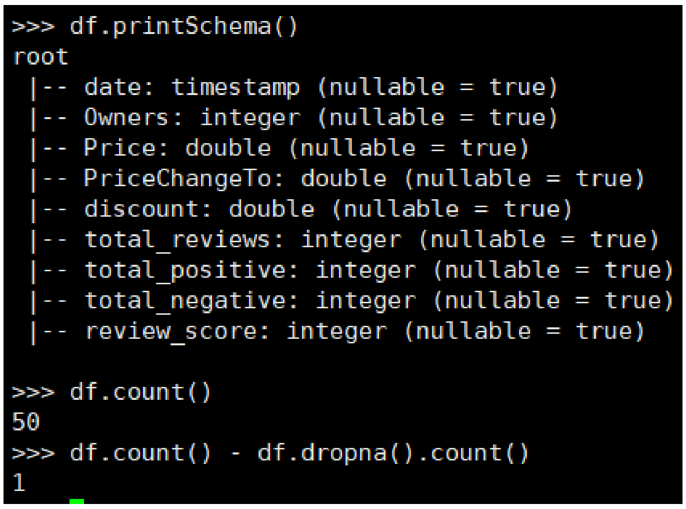
```
Using Python version 2.7.5 (default, Sep 26 2019 13:23:47)
SparkSession available as 'spark'.
>>> from pyspark.sql import SparkSession
>>> from pyspark import SparkContext
>>> import pyspark.sql.functions as func
>>> from pyspark.sql import Window
>>> from pyspark.sql.types import *
>>>
>>> sc = SparkContext(appName="discount")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/opt/cloudera/parcels/CDH-6.3.4-1.cdh6.3.4.p0.6626826/lib/spark/python/pyspark/context.py", line 120, in __in
    SparkContext._ensure_initialized(self, gateway=gateway, conf=conf)
  File "/opt/cloudera/parcels/CDH-6.3.4-1.cdh6.3.4.p0.6626826/lib/spark/python/pyspark/context.py", line 319, in _ens
    callsite.function, callsite.file, callsite.linenum))
ValueError: Cannot run multiple SparkContexts at once; existing SparkContext(app=PySparkShell, master=yarn) created b
/shell.py:41
>>> sc = SparkContext.getOrCreate(appName="discount")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: getOrCreate() got an unexpected keyword argument 'appName'
>>> sc = SparkContext.getOrCreate()
>>> spark = SparkSession(sc)
>>> hdfs_path = "./project/discount/"
>>> input_path = hdfs_path + "raw_input/"
>>> df = spark.read.csv(input_path+"240.csv", header=True, inferSchema=True)
>>> df.show(10)
+-------------------+--------+-----+-------------+--------+-------------+--------------+--------------+------------+
|               date|  Owners|Price|PriceChangeTo|discount|total_reviews|total_positive|total_negative|review_score|
+-------------------+--------+-----+-------------+--------+-------------+--------------+--------------+------------+
|2015-04-01 00:00:00|13698000|  0.0|        19.99|    null|        34324|         33346|           978|           9|
|2015-06-11 00:00:00|13830000|19.99|         7.49|   37.47|        36287|         35177|          1110|           9|
|2015-06-22 00:00:00|13893000| 7.49|        19.99|  266.89|        36806|         35677|          1129|           9|
|2015-11-25 00:00:00|14300000|19.99|         7.49|   37.47|        40673|         39321|          1352|           9|
|2015-12-01 00:00:00|14314000| 7.49|        19.99|  266.89|        40938|         39569|          1369|           9|
|2015-12-22 00:00:00|14362000|19.99|         7.49|   37.47|        41547|         40150|          1397|           9|
|2016-01-04 00:00:00|14444000| 7.49|        19.99|  266.89|        42245|         40806|          1439|           9|
|2016-02-05 00:00:00|15218000|19.99|         4.99|   24.96|        43308|         41807|          1501|           9|
|2016-02-12 00:00:00|14906000| 4.99|        19.99|   400.6|        43706|         42178|          1528|           9|
|2016-11-23 00:00:00|15629000|19.99|         6.79|   33.97|        50195|         48312|          1883|           9|
+-------------------+--------+-----+-------------+--------+-------------+--------------+--------------+------------+
```

2. Check schema and rows

```
df.printSchema()
df.count()
```

3. Check how many null values drop if drop null.

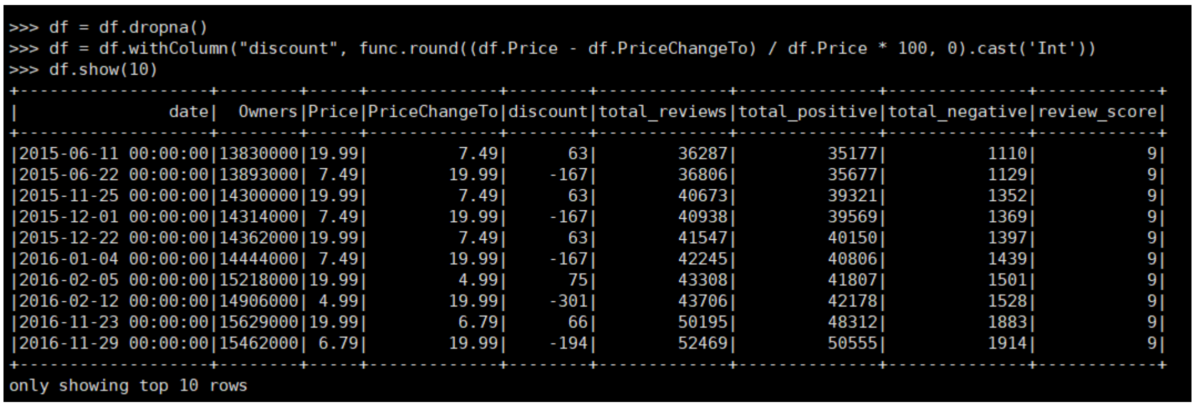
```
df.count() - df.dropna().count()
```



```
>>> df.printSchema()
root
 |-- date: timestamp (nullable = true)
 |-- Owners: integer (nullable = true)
 |-- Price: double (nullable = true)
 |-- PriceChangeTo: double (nullable = true)
 |-- discount: double (nullable = true)
 |-- total_reviews: integer (nullable = true)
 |-- total_positive: integer (nullable = true)
 |-- total_negative: integer (nullable = true)
 |-- review_score: integer (nullable = true)

>>> df.count()
50
>>> df.count() - df.dropna().count()
1
```

4. Found that discount calculation is Wrong, recalculate discount

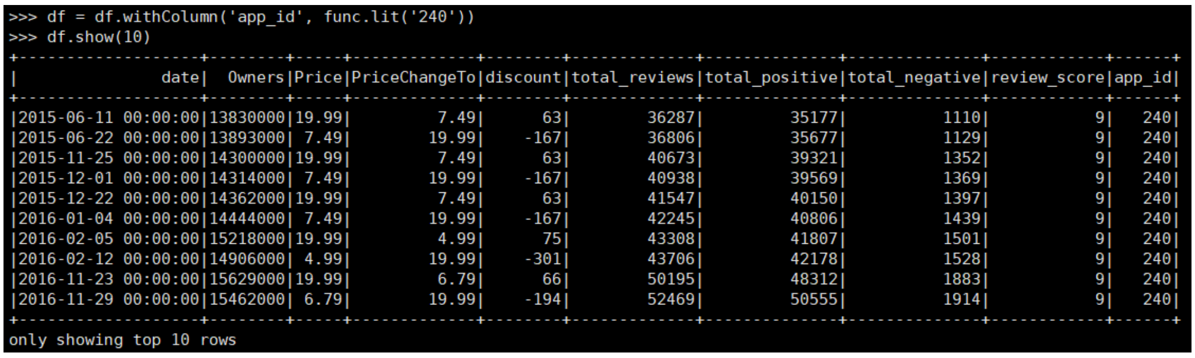
```
df = df.dropna()
df = df.withColumn("discount", func.round((df.Price - df.PriceChangeTo) /
df.Price * 100, 0).cast('Int'))
df.show(10)
```



```
>>> df = df.dropna()
>>> df = df.withColumn("discount", func.round((df.Price - df.PriceChangeTo) / df.Price * 100, 0).cast('Int'))
>>> df.show(10)
+-------------------+--------+-----+-------------+--------+-------------+--------------+--------------+------------+
|               date|  Owners|Price|PriceChangeTo|discount|total_reviews|total_positive|total_negative|review_score|
+-------------------+--------+-----+-------------+--------+-------------+--------------+--------------+------------+
|2015-06-11 00:00:00|13830000|19.99|         7.49|      63|        36287|         35177|          1110|           9|
|2015-06-22 00:00:00|13893000| 7.49|        19.99|    -167|        36806|         35677|          1129|           9|
|2015-11-25 00:00:00|14300000|19.99|         7.49|      63|        40673|         39321|          1352|           9|
|2015-12-01 00:00:00|14314000| 7.49|        19.99|    -167|        40938|         39569|          1369|           9|
|2015-12-22 00:00:00|14362000|19.99|         7.49|      63|        41547|         40150|          1397|           9|
|2016-01-04 00:00:00|14444000| 7.49|        19.99|    -167|        42245|         40806|          1439|           9|
|2016-02-05 00:00:00|15218000|19.99|         4.99|      75|        43308|         41807|          1501|           9|
|2016-02-12 00:00:00|14906000| 4.99|        19.99|    -301|        43706|         42178|          1528|           9|
|2016-11-23 00:00:00|15629000|19.99|         6.79|      66|        50195|         48312|          1883|           9|
|2016-11-29 00:00:00|15462000| 6.79|        19.99|    -194|        52469|         50555|          1914|           9|
+-------------------+--------+-----+-------------+--------+-------------+--------------+--------------+------------+
only showing top 10 rows
```

5. Add app id column, for future combine use.

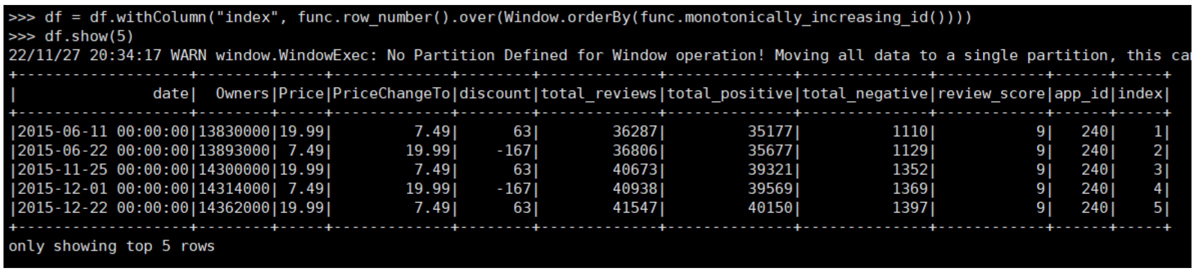
```
df = df.withColumn('app_id', func.lit('240'))
df.show(10)
```



```
>>> df = df.withColumn('app_id', func.lit('240'))
>>> df.show(10)
+-------------------+--------+-----+-------------+--------+-------------+--------------+--------------+------------+------+
|               date|  Owners|Price|PriceChangeTo|discount|total_reviews|total_positive|total_negative|review_score|app_id|
+-------------------+--------+-----+-------------+--------+-------------+--------------+--------------+------------+------+
|2015-06-11 00:00:00|13830000|19.99|         7.49|      63|        36287|         35177|          1110|           9|   240|
|2015-06-22 00:00:00|13893000| 7.49|        19.99|    -167|        36806|         35677|          1129|           9|   240|
|2015-11-25 00:00:00|14300000|19.99|         7.49|      63|        40673|         39321|          1352|           9|   240|
|2015-12-01 00:00:00|14314000| 7.49|        19.99|    -167|        40938|         39569|          1369|           9|   240|
|2015-12-22 00:00:00|14362000|19.99|         7.49|      63|        41547|         40150|          1397|           9|   240|
|2016-01-04 00:00:00|14444000| 7.49|        19.99|    -167|        42245|         40806|          1439|           9|   240|
|2016-02-05 00:00:00|15218000|19.99|         4.99|      75|        43308|         41807|          1501|           9|   240|
|2016-02-12 00:00:00|14906000| 4.99|        19.99|    -301|        43706|         42178|          1528|           9|   240|
|2016-11-23 00:00:00|15629000|19.99|         6.79|      66|        50195|         48312|          1883|           9|   240|
|2016-11-29 00:00:00|15462000| 6.79|        19.99|    -194|        52469|         50555|          1914|           9|   240|
+-------------------+--------+-----+-------------+--------+-------------+--------------+--------------+------------+------+
only showing top 10 rows
```

6. Add index column

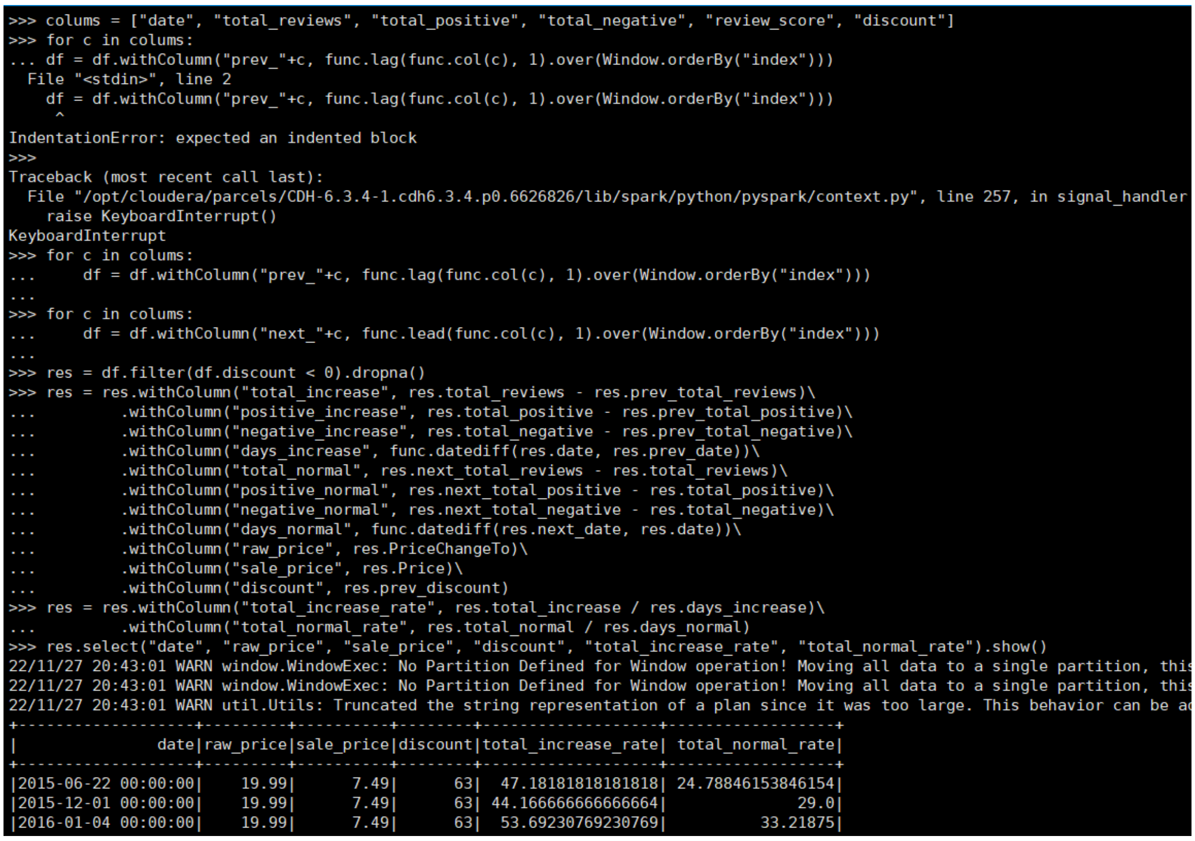
```
df = df.withColumn("index",
func.row_number().over(Window.orderBy(func.monotonically_increasing_id())))
df.show(5)
```



```
>>> df = df.withColumn("index", func.row_number().over(Window.orderBy(func.monotonically_increasing_id())))
>>> df.show(5)
22/11/27 20:34:17 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this ca
+-------------------+--------+-----+------------+--------+-------------+--------------+--------------+------------+------+-----+
|               date|  Owners|Price|PriceChangeTo|discount|total_reviews|total_positive|total_negative|review_score|app_id|index|
+-------------------+--------+-----+------------+--------+-------------+--------------+--------------+------------+------+-----+
|2015-06-11 00:00:00|13830000|19.99|        7.49|      63|        36287|         35177|          1110|           9|   240|    1|
|2015-06-22 00:00:00|13893000| 7.49|       19.99|    -167|        36806|         35677|          1129|           9|   240|    2|
|2015-11-25 00:00:00|14300000|19.99|        7.49|      63|        40673|         39321|          1352|           9|   240|    3|
|2015-12-01 00:00:00|14314000| 7.49|       19.99|    -167|        40938|         39569|          1369|           9|   240|    4|
|2015-12-22 00:00:00|14362000|19.99|        7.49|      63|        41547|         40150|          1397|           9|   240|    5|
+-------------------+--------+-----+------------+--------+-------------+--------------+--------------+------------+------+-----+
only showing top 5 rows
```

7. Bunch of codes to create desired features, Full version in ETL folders as single .py Job

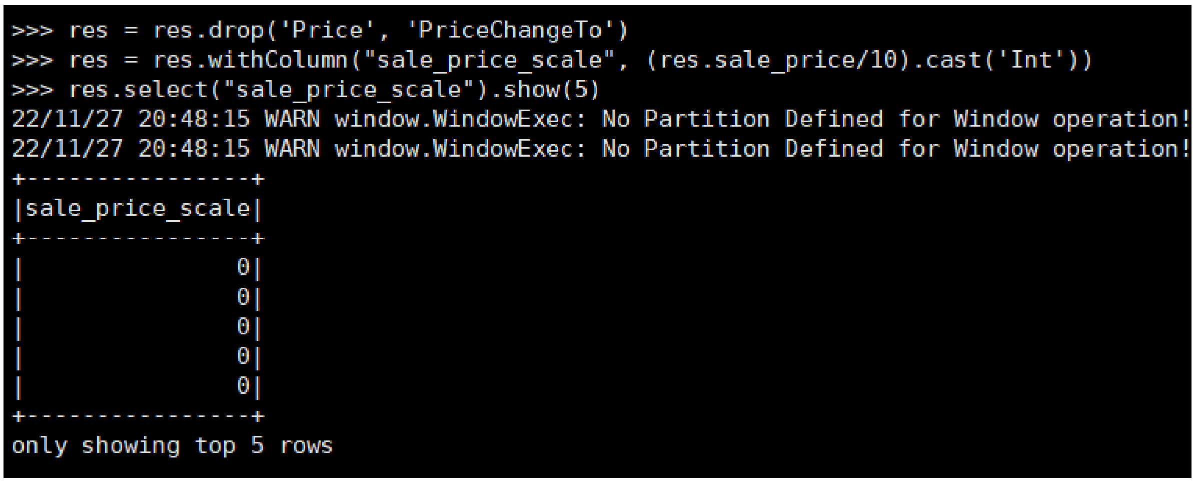
```
colums = ["date", "total_reviews", "total_positive", "total_negative",
"review_score", "discount"]
for c in colums:
    df = df.withColumn("prev_"+c, func.lag(func.col(c),
1).over(Window.orderBy("index")))
for c in colums:
    df = df.withColumn("next_"+c, func.lead(func.col(c),
1).over(Window.orderBy("index")))
res = df.filter(df.discount < 0).dropna()
res = res.withColumn("total_increase", res.total_reviews -
res.prev_total_reviews)\
        .withColumn("positive_increase", res.total_positive -
res.prev_total_positive)\
        .withColumn("negative_increase", res.total_negative -
res.prev_total_negative)\
        .withColumn("days_increase", func.datediff(res.date, res.prev_date))\
        .withColumn("total_normal", res.next_total_reviews - res.total_reviews)\
        .withColumn("positive_normal", res.next_total_positive -
res.total_positive)\
        .withColumn("negative_normal", res.next_total_negative -
res.total_negative)\
        .withColumn("days_normal", func.datediff(res.next_date, res.date))\
        .withColumn("raw_price", res.PriceChangeTo)\
        .withColumn("sale_price", res.Price)\
        .withColumn("discount", res.prev_discount)
res = res.withColumn("total_increase_rate", res.total_increase /
res.days_increase)\
        .withColumn("total_normal_rate", res.total_normal / res.days_normal)
res.select("date", "raw_price", "sale_price", "discount", "total_increase_rate",
"total_normal_rate").show()
```



```
>>> colums = ["date", "total_reviews", "total_positive", "total_negative", "review_score", "discount"]
>>> for c in colums:
... df = df.withColumn("prev_"+c, func.lag(func.col(c), 1).over(Window.orderBy("index")))
  File "<stdin>", line 2
    df = df.withColumn("prev_"+c, func.lag(func.col(c), 1).over(Window.orderBy("index")))
     ^
IndentationError: expected an indented block
>>>
Traceback (most recent call last):
  File "/opt/cloudera/parcels/CDH-6.3.4-1.cdh6.3.4.p0.6626826/lib/spark/python/pyspark/context.py", line 257, in signal_handler
    raise KeyboardInterrupt()
KeyboardInterrupt
>>> for c in colums:
...     df = df.withColumn("prev_"+c, func.lag(func.col(c), 1).over(Window.orderBy("index")))
...
>>> for c in colums:
...     df = df.withColumn("next_"+c, func.lead(func.col(c), 1).over(Window.orderBy("index")))
...
>>> res = df.filter(df.discount < 0).dropna()
>>> res = res.withColumn("total_increase", res.total_reviews - res.prev_total_reviews)\
...         .withColumn("positive_increase", res.total_positive - res.prev_total_positive)\
...         .withColumn("negative_increase", res.total_negative - res.prev_total_negative)\
...         .withColumn("days_increase", func.datediff(res.date, res.prev_date))\
...         .withColumn("total_normal", res.next_total_reviews - res.total_reviews)\
...         .withColumn("positive_normal", res.next_total_positive - res.total_positive)\
...         .withColumn("negative_normal", res.next_total_negative - res.total_negative)\
...         .withColumn("days_normal", func.datediff(res.next_date, res.date))\
...         .withColumn("raw_price", res.PriceChangeTo)\
...         .withColumn("sale_price", res.Price)\
...         .withColumn("discount", res.prev_discount)
>>> res = res.withColumn("total_increase_rate", res.total_increase / res.days_increase)\
...         .withColumn("total_normal_rate", res.total_normal / res.days_normal)
>>> res.select("date", "raw_price", "sale_price", "discount", "total_increase_rate", "total_normal_rate").show()
22/11/27 20:43:01 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this
22/11/27 20:43:01 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this
22/11/27 20:43:01 WARN util.Utils: Truncated the string representation of a plan since it was too large. This behavior can be ad
+-------------------+---------+----------+--------+-------------------+------------------+
|               date|raw_price|sale_price|discount|total_increase_rate| total_normal_rate|
+-------------------+---------+----------+--------+-------------------+------------------+
|2015-06-22 00:00:00|    19.99|      7.49|      63|  47.18181818181818| 24.78846153846154|
|2015-12-01 00:00:00|    19.99|      7.49|      63| 44.166666666666664|              29.0|
|2016-01-04 00:00:00|    19.99|      7.49|      63|  53.69230769230769|          33.21875|
```

8. Drop cols don't need, calculate sale_price_scale

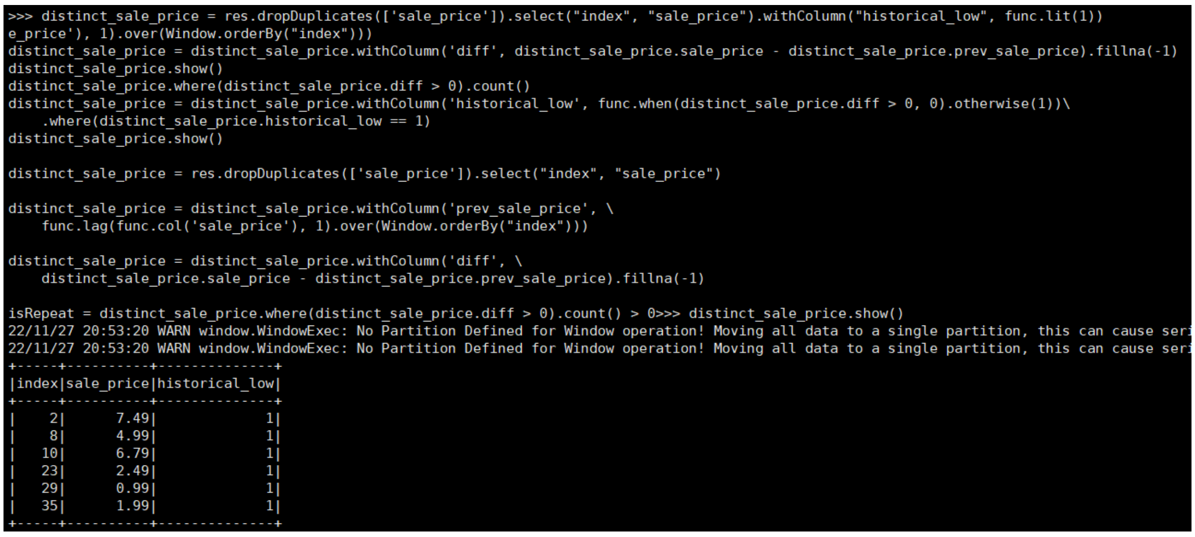
```
res = res.drop('Price', 'PriceChangeTo')
res = res.withColumn("sale_price_scale", (res.sale_price/10).cast('Int'))
res.select("sale_price_scale").show(5)
```



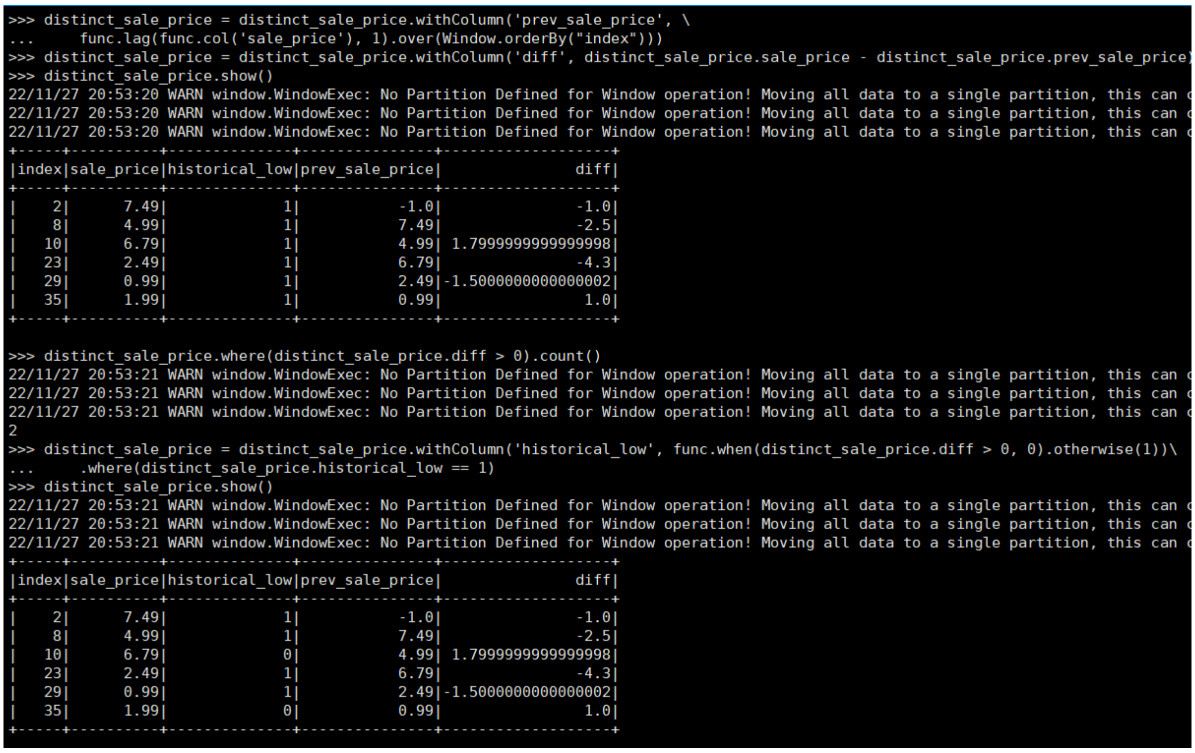
```
>>> res = res.drop('Price', 'PriceChangeTo')
>>> res = res.withColumn("sale_price_scale", (res.sale_price/10).cast('Int'))
>>> res.select("sale_price_scale").show(5)
22/11/27 20:48:15 WARN window.WindowExec: No Partition Defined for Window operation!
22/11/27 20:48:15 WARN window.WindowExec: No Partition Defined for Window operation!
+----------------+
|sale_price_scale|
+----------------+
|               0|
|               0|
|               0|
|               0|
|               0|
+----------------+
only showing top 5 rows
```

9. Steps to get Historical Lowest Price Label, Full version in ETL folder single *.py Job

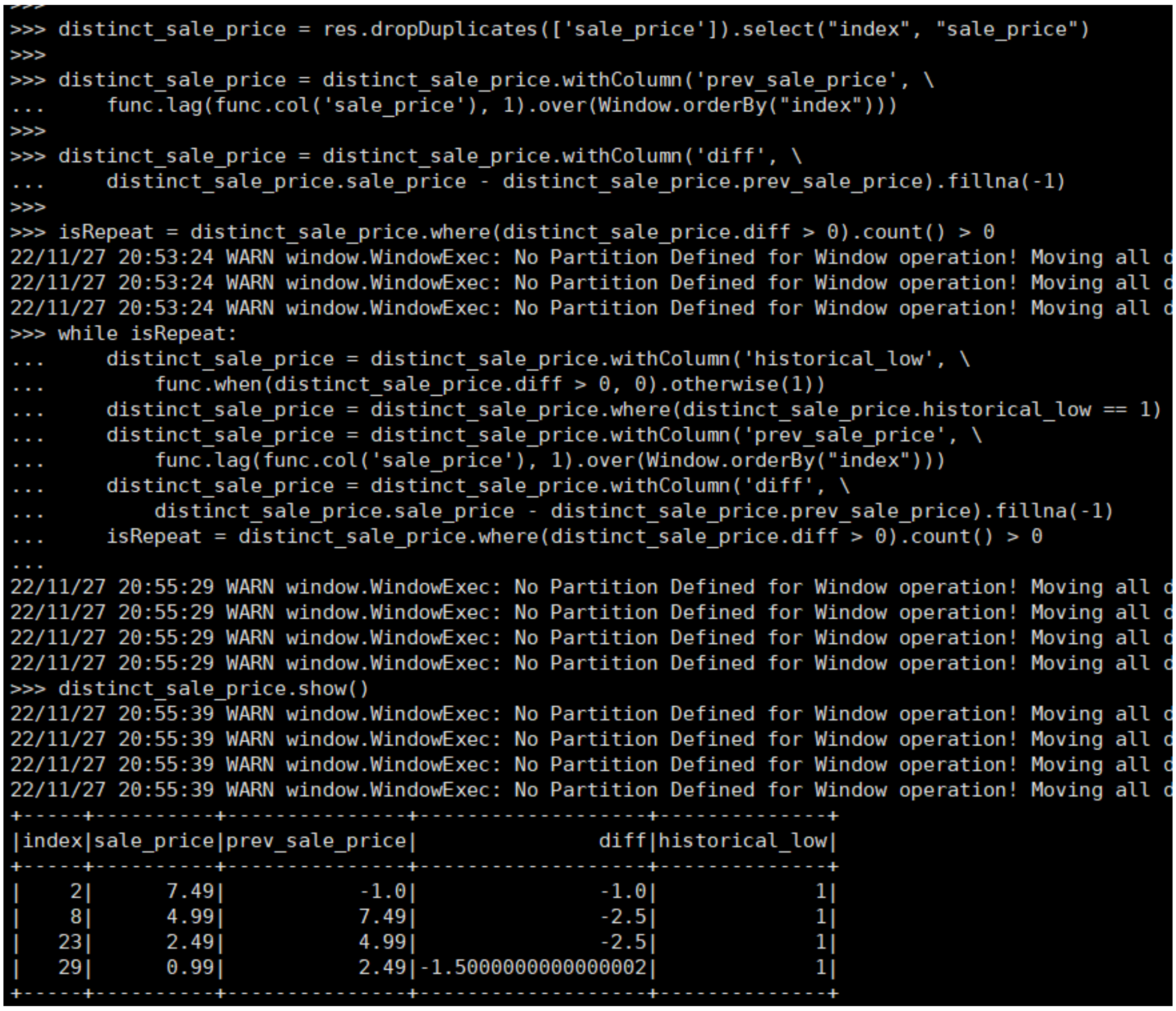
```
distinct_sale_price = res.dropDuplicates(['sale_price']).select("index",
"sale_price").withColumn("historical_low", func.lit(1))
distinct_sale_price.show()
distinct_sale_price = distinct_sale_price.withColumn('prev_sale_price', \
    func.lag(func.col('sale_price'), 1).over(Window.orderBy("index")))
distinct_sale_price = distinct_sale_price.withColumn('diff',
distinct_sale_price.sale_price - distinct_sale_price.prev_sale_price).fillna(-1)
distinct_sale_price.show()
distinct_sale_price.where(distinct_sale_price.diff > 0).count()
distinct_sale_price = distinct_sale_price.withColumn('historical_low',
func.when(distinct_sale_price.diff > 0, 0).otherwise(1))\
    .where(distinct_sale_price.historical_low == 1)
distinct_sale_price.show()

distinct_sale_price = res.dropDuplicates(['sale_price']).select("index",
"sale_price")

distinct_sale_price = distinct_sale_price.withColumn('prev_sale_price', \
    func.lag(func.col('sale_price'), 1).over(Window.orderBy("index")))

distinct_sale_price = distinct_sale_price.withColumn('diff', \
    distinct_sale_price.sale_price -
distinct_sale_price.prev_sale_price).fillna(-1)

isRepeat = distinct_sale_price.where(distinct_sale_price.diff > 0).count() > 0

while isRepeat:
    distinct_sale_price = distinct_sale_price.withColumn('historical_low', \
        func.when(distinct_sale_price.diff > 0, 0).otherwise(1))
    distinct_sale_price =
distinct_sale_price.where(distinct_sale_price.historical_low == 1)

    distinct_sale_price = distinct_sale_price.withColumn('prev_sale_price', \
        func.lag(func.col('sale_price'), 1).over(Window.orderBy("index")))

    distinct_sale_price = distinct_sale_price.withColumn('diff', \
        distinct_sale_price.sale_price -
distinct_sale_price.prev_sale_price).fillna(-1)

    isRepeat = distinct_sale_price.where(distinct_sale_price.diff > 0).count() >
0

distinct_sale_price.show()
```



```
>>> distinct_sale_price = res.dropDuplicates(['sale_price']).select("index", "sale_price").withColumn("historical_low", func.lit(1))
e_price'), 1).over(Window.orderBy("index")))
distinct_sale_price = distinct_sale_price.withColumn('diff', distinct_sale_price.sale_price - distinct_sale_price.prev_sale_price).fillna(-1)
distinct_sale_price.show()
distinct_sale_price.where(distinct_sale_price.diff > 0).count()
distinct_sale_price = distinct_sale_price.withColumn('historical_low', func.when(distinct_sale_price.diff > 0, 0).otherwise(1))\
    .where(distinct_sale_price.historical_low == 1)
distinct_sale_price.show()

distinct_sale_price = res.dropDuplicates(['sale_price']).select("index", "sale_price")

distinct_sale_price = distinct_sale_price.withColumn('prev_sale_price', \
    func.lag(func.col('sale_price'), 1).over(Window.orderBy("index")))

distinct_sale_price = distinct_sale_price.withColumn('diff', \
    distinct_sale_price.sale_price - distinct_sale_price.prev_sale_price).fillna(-1)

isRepeat = distinct_sale_price.where(distinct_sale_price.diff > 0).count() > 0>>> distinct_sale_price.show()
22/11/27 20:53:20 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause seri
22/11/27 20:53:20 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause seri
+-----+----------+--------------+
|index|sale_price|historical_low|
+-----+----------+--------------+
|    2|      7.49|             1|
|    8|      4.99|             1|
|   10|      6.79|             1|
|   23|      2.49|             1|
|   29|      0.99|             1|
|   35|      1.99|             1|
+-----+----------+--------------+
```



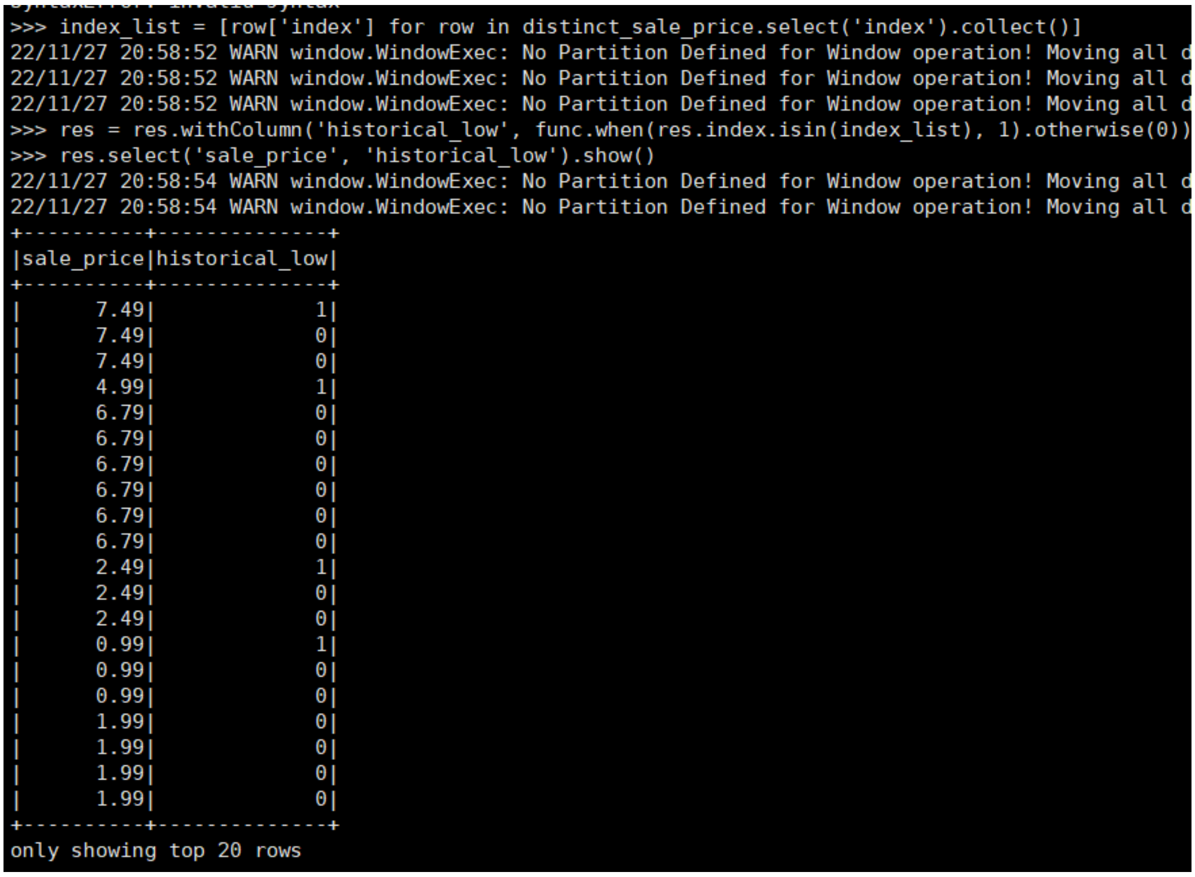
```
>>> distinct_sale_price = distinct_sale_price.withColumn('prev_sale_price', \
...     func.lag(func.col('sale_price'), 1).over(Window.orderBy("index")))
>>> distinct_sale_price = distinct_sale_price.withColumn('diff', distinct_sale_price.sale_price - distinct_sale_price.prev_sale_price)
>>> distinct_sale_price.show()
22/11/27 20:53:20 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can c
22/11/27 20:53:20 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can c
22/11/27 20:53:20 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can c
+-----+----------+--------------+---------------+-------------------+
|index|sale_price|historical_low|prev_sale_price|               diff|
+-----+----------+--------------+---------------+-------------------+
|    2|      7.49|             1|           -1.0|               -1.0|
|    8|      4.99|             1|           7.49|               -2.5|
|   10|      6.79|             1|           4.99| 1.7999999999999998|
|   23|      2.49|             1|           6.79|               -4.3|
|   29|      0.99|             1|           2.49|-1.5000000000000002|
|   35|      1.99|             1|           0.99|                1.0|
+-----+----------+--------------+---------------+-------------------+

>>> distinct_sale_price.where(distinct_sale_price.diff > 0).count()
22/11/27 20:53:21 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can c
22/11/27 20:53:21 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can c
22/11/27 20:53:21 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can c
2
>>> distinct_sale_price = distinct_sale_price.withColumn('historical_low', func.when(distinct_sale_price.diff > 0, 0).otherwise(1))\
...     .where(distinct_sale_price.historical_low == 1)
>>> distinct_sale_price.show()
22/11/27 20:53:21 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can c
22/11/27 20:53:21 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can c
22/11/27 20:53:21 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can c
+-----+----------+--------------+---------------+-------------------+
|index|sale_price|historical_low|prev_sale_price|               diff|
+-----+----------+--------------+---------------+-------------------+
|    2|      7.49|             1|           -1.0|               -1.0|
|    8|      4.99|             1|           7.49|               -2.5|
|   10|      6.79|             0|           4.99| 1.7999999999999998|
|   23|      2.49|             1|           6.79|               -4.3|
|   29|      0.99|             1|           2.49|-1.5000000000000002|
|   35|      1.99|             0|           0.99|                1.0|
+-----+----------+--------------+---------------+-------------------+
```



```
>>> distinct_sale_price = res.dropDuplicates(['sale_price']).select("index", "sale_price")
>>>
>>> distinct_sale_price = distinct_sale_price.withColumn('prev_sale_price', \
...     func.lag(func.col('sale_price'), 1).over(Window.orderBy("index")))
>>>
>>> distinct_sale_price = distinct_sale_price.withColumn('diff', \
...     distinct_sale_price.sale_price - distinct_sale_price.prev_sale_price).fillna(-1)
>>>
>>> isRepeat = distinct_sale_price.where(distinct_sale_price.diff > 0).count() > 0
22/11/27 20:53:24 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
22/11/27 20:53:24 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
22/11/27 20:53:24 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
>>> while isRepeat:
...     distinct_sale_price = distinct_sale_price.withColumn('historical_low', \
...         func.when(distinct_sale_price.diff > 0, 0).otherwise(1))
...     distinct_sale_price = distinct_sale_price.where(distinct_sale_price.historical_low == 1)
...     distinct_sale_price = distinct_sale_price.withColumn('prev_sale_price', \
...         func.lag(func.col('sale_price'), 1).over(Window.orderBy("index")))
...     distinct_sale_price = distinct_sale_price.withColumn('diff', \
...         distinct_sale_price.sale_price - distinct_sale_price.prev_sale_price).fillna(-1)
...     isRepeat = distinct_sale_price.where(distinct_sale_price.diff > 0).count() > 0
...
22/11/27 20:55:29 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
22/11/27 20:55:29 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
22/11/27 20:55:29 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
22/11/27 20:55:29 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
>>> distinct_sale_price.show()
22/11/27 20:55:39 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
22/11/27 20:55:39 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
22/11/27 20:55:39 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
22/11/27 20:55:39 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
+-----+----------+---------------+-------------------+--------------+
|index|sale_price|prev_sale_price|               diff|historical_low|
+-----+----------+---------------+-------------------+--------------+
|    2|      7.49|           -1.0|               -1.0|             1|
|    8|      4.99|           7.49|               -2.5|             1|
|   23|      2.49|           4.99|               -2.5|             1|
|   29|      0.99|           2.49|-1.5000000000000002|             1|
+-----+----------+---------------+-------------------+--------------+
```

10. Joint back to main df

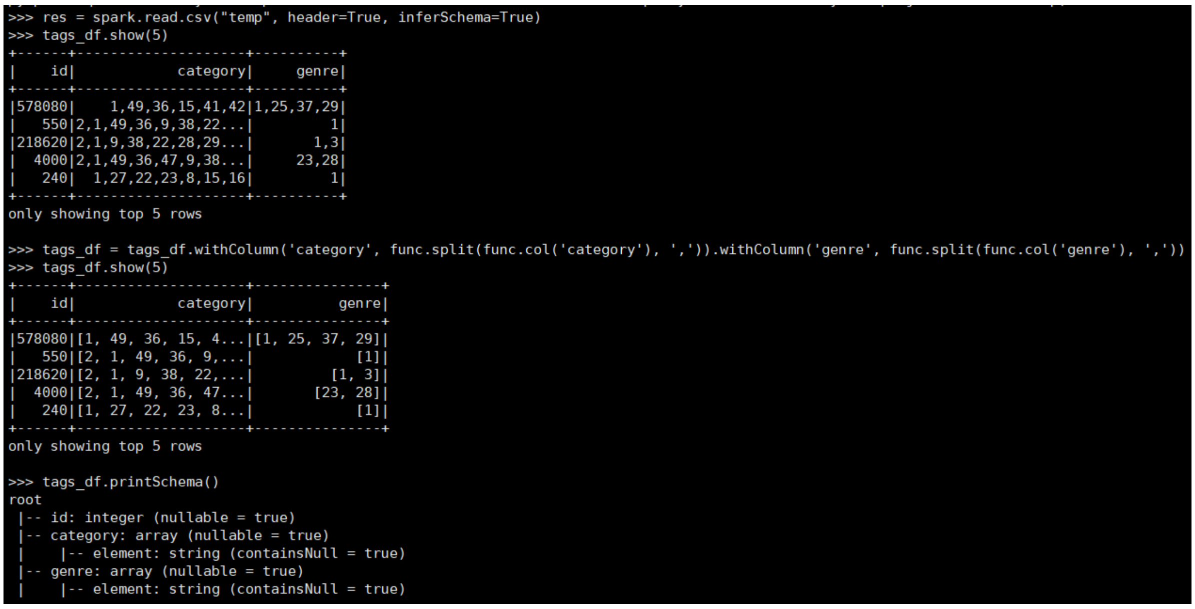
```
index_list = [row['index'] for row in
distinct_sale_price.select('index').collect()]
res = res.withColumn('historical_low', func.when(res.index.isin(index_list),
1).otherwise(0))
res.select('sale_price', 'historical_low').show()
```



```
>>> index_list = [row['index'] for row in distinct_sale_price.select('index').collect()]
22/11/27 20:58:52 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
22/11/27 20:58:52 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
22/11/27 20:58:52 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
>>> res = res.withColumn('historical_low', func.when(res.index.isin(index_list), 1).otherwise(0))
>>> res.select('sale_price', 'historical_low').show()
22/11/27 20:58:54 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
22/11/27 20:58:54 WARN window.WindowExec: No Partition Defined for Window operation! Moving all d
+----------+--------------+
|sale_price|historical_low|
+----------+--------------+
|      7.49|             1|
|      7.49|             0|
|      7.49|             0|
|      4.99|             1|
|      6.79|             0|
|      6.79|             0|
|      6.79|             0|
|      6.79|             0|
|      6.79|             0|
|      6.79|             0|
|      2.49|             1|
|      2.49|             0|
|      2.49|             0|
|      0.99|             1|
|      0.99|             0|
|      0.99|             0|
|      1.99|             0|
|      1.99|             0|
|      1.99|             0|
|      1.99|             0|
+----------+--------------+
only showing top 20 rows
```

11. Load Category and Genre Dataset

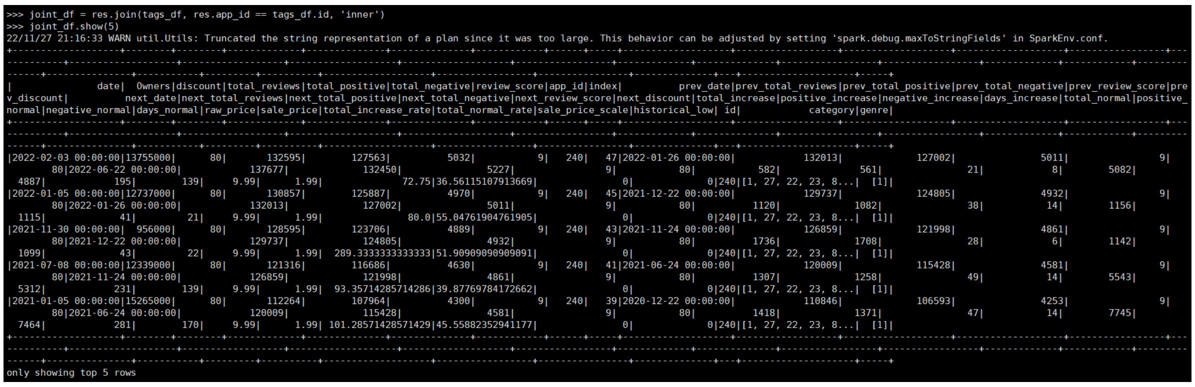
```
tags_df = spark.read.csv(hdfs_path+"tags_input/joint_category_genre.csv",
header=True, inferSchema=True)
tags_df.show(5)
tags_df = tags_df.withColumn('category', func.split(func.col('category'),
',')).withColumn('genre', func.split(func.col('genre'), ','))
tags_df.show(5)
tags_df.printSchema()
```



```
>>> res = spark.read.csv("temp", header=True, inferSchema=True)
>>> tags_df.show(5)
+------+--------------------+----------+
|    id|            category|     genre|
+------+--------------------+----------+
|578080|    1,49,36,15,41,42|1,25,37,29|
|   550|2,1,49,36,9,38,22...|         1|
|218620|2,1,9,38,22,28,29...|       1,3|
|  4000|2,1,49,36,47,9,38...|     23,28|
|   240|  1,27,22,23,8,15,16|         1|
+------+--------------------+----------+
only showing top 5 rows

>>> tags_df = tags_df.withColumn('category', func.split(func.col('category'), ',')).withColumn('genre', func.split(func.col('genre'), ','))
>>> tags_df.show(5)
+------+--------------------+--------------+
|    id|            category|         genre|
+------+--------------------+--------------+
|578080|[1, 49, 36, 15, 4...|[1, 25, 37, 29]|
|   550|[2, 1, 49, 36, 9,...|           [1]|
|218620|[2, 1, 9, 38, 22,...|        [1, 3]|
|  4000|[2, 1, 49, 36, 47...|      [23, 28]|
|   240|[1, 27, 22, 23, 8...|           [1]|
+------+--------------------+--------------+
only showing top 5 rows

>>> tags_df.printSchema()
root
 |-- id: integer (nullable = true)
 |-- category: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- genre: array (nullable = true)
 |    |-- element: string (containsNull = true)
```

12. Inner joint with main df

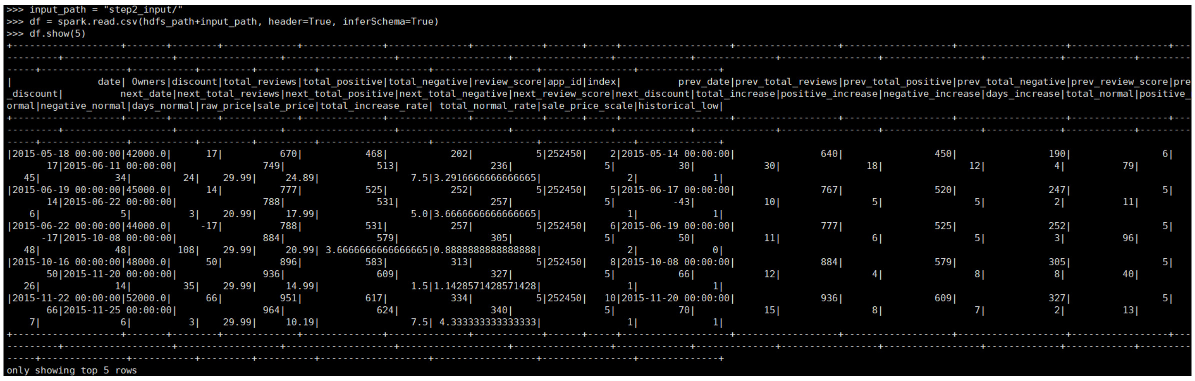
```
joint_df = res.join(tags_df, res.app_id == tags_df.id, 'inner')
joint_df.show(5)
```



```
>>> joint_df = res.join(tags_df, res.app_id == tags_df.id, 'inner')
>>> joint_df.show(5)
22/11/27 21:16:33 WARN util.Utils: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.debug.maxToStringFields' in SparkEnv.conf.
+-------------------+--------+--------+-------------+--------------+--------------+------------+------+-----+-------------------+------------------+-------------------+-------------------+-----------------+---
----------+-------------------+------------------+-------------------+-------------------+-----------------+-------------+--------------+-----------------+-----------------+-------------+------------+---------
-----+---------------+-----------+---------+----------+-------------------+-----------------+----------------+--------------+---+--------------------+-----+
|              date|  Owners|discount|total_reviews|total_positive|total_negative|review_score|app_id|index|          prev_date|prev_total_reviews|prev_total_positive|prev_total_negative|prev_review_score|pre
v_discount|          next_date|next_total_reviews|next_total_positive|next_total_negative|next_review_score|next_discount|total_increase|positive_increase|negative_increase|days_increase|total_normal|positive_
normal|negative_normal|days_normal|raw_price|sale_price|total_increase_rate|total_normal_rate|sale_price_scale|historical_low| id|           category|genre|
+-------------------+--------+--------+-------------+--------------+--------------+------------+------+-----+-------------------+------------------+-------------------+-------------------+-----------------+---
----------+-------------------+------------------+-------------------+-------------------+-----------------+-------------+--------------+-----------------+-----------------+-------------+------------+---------
-----+---------------+-----------+---------+----------+-------------------+-----------------+----------------+--------------+---+--------------------+-----+
|2022-02-03 00:00:00|13755000|      80|       132595|        127563|          5032|           9|   240|   47|2022-01-26 00:00:00|            132013|             127002|               5011|                9|
        80|2022-06-22 00:00:00|            137677|             132450|               5227|                9|           80|           582|              561|               21|            8|        5082|
  4887|            195|        139|     9.99|      1.99|              72.75|36.56115107913669|               0|             0|240|[1, 27, 22, 23, 8...|  [1]|
|2022-01-05 00:00:00|12737000|      80|       130857|        125887|          4970|           9|   240|   45|2021-12-22 00:00:00|            129737|             124805|               4932|                9|
        80|2022-01-26 00:00:00|            132013|             127002|               5011|                9|           80|          1120|             1082|               38|           14|        1156|
  1115|             41|         21|     9.99|      1.99|               80.0|55.04761904761905|               0|             0|240|[1, 27, 22, 23, 8...|  [1]|
|2021-11-30 00:00:00|  956000|      80|       128595|        123706|          4889|           9|   240|   43|2021-11-24 00:00:00|            126859|             121998|               4861|                9|
        80|2021-12-22 00:00:00|            129737|             124805|               4932|                9|           80|          1736|             1708|               28|            6|        1142|
  1099|             43|         22|     9.99|      1.99|  289.3333333333333|51.90909090909091|               0|             0|240|[1, 27, 22, 23, 8...|  [1]|
|2021-07-08 00:00:00|12339000|      80|       121316|        116686|          4630|           9|   240|   41|2021-06-24 00:00:00|            120009|             115428|               4581|                9|
        80|2021-11-24 00:00:00|            126859|             121998|               4861|                9|           80|          1307|             1258|               49|           14|        5543|
  5312|            231|        139|     9.99|      1.99|  93.35714285714286|39.87769784172662|               0|             0|240|[1, 27, 22, 23, 8...|  [1]|
|2021-01-05 00:00:00|15265000|      80|       112264|        107964|          4300|           9|   240|   39|2020-12-22 00:00:00|            110846|             106593|               4253|                9|
        80|2021-06-24 00:00:00|            120009|             115428|               4581|                9|           80|          1418|             1371|               47|           14|        7745|
  7464|            281|        170|     9.99|      1.99| 101.28571428571429|45.55882352941177|               0|             0|240|[1, 27, 22, 23, 8...|  [1]|
+-------------------+--------+--------+-------------+--------------+--------------+------------+------+-----+-------------------+------------------+-------------------+-------------------+-----------------+---
----------+-------------------+------------------+-------------------+-------------------+-----------------+-------------+--------------+-----------------+-----------------+-------------+------------+---------
-----+---------------+-----------+---------+----------+-------------------+-----------------+----------------+--------------+---+--------------------+-----+
only showing top 5 rows
```

13. Prepare Dataset for analysis

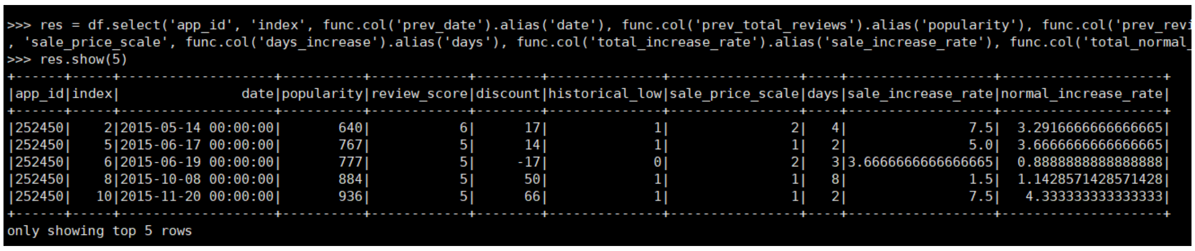
```
input_path = "step2_input/"
df = spark.read.csv(hdfs_path+input_path, header=True, inferSchema=True)
df.show(5)
```



```
>>> input_path = "step2_input/"
>>> df = spark.read.csv(hdfs_path+input_path, header=True, inferSchema=True)
>>> df.show(5)
+-------------------+-------+--------+-------------+--------------+--------------+------------+------+-----+-------------------+------------------+-------------------+-------------------+-----------------+---
---------+-------------------+------------------+-------------------+-------------------+-----------------+-------------+--------------+-----------------+-----------------+-------------+------------+---------
-----+---------------+-----------+---------+----------+-------------------+------------------+----------------+--------------+
|              date| Owners|discount|total_reviews|total_positive|total_negative|review_score|app_id|index|          prev_date|prev_total_reviews|prev_total_positive|prev_total_negative|prev_review_score|pre
_discount|          next_date|next_total_reviews|next_total_positive|next_total_negative|next_review_score|next_discount|total_increase|positive_increase|negative_increase|days_increase|total_normal|positive_
ormal|negative_normal|days_normal|raw_price|sale_price|total_increase_rate| total_normal_rate|sale_price_scale|historical_low|
+-------------------+-------+--------+-------------+--------------+--------------+------------+------+-----+-------------------+------------------+-------------------+-------------------+-----------------+---
---------+-------------------+------------------+-------------------+-------------------+-----------------+-------------+--------------+-----------------+-----------------+-------------+------------+---------
-----+---------------+-----------+---------+----------+-------------------+------------------+----------------+--------------+
|2015-05-18 00:00:00|42000.0|      17|          670|           468|           202|           5|252450|    2|2015-05-14 00:00:00|               640|                450|                190|                6|
       17|2015-06-11 00:00:00|               749|                513|                236|                5|            5|            30|               30|               18|           12|           4|          79|
    45|             34|         24|    29.99|     24.89|                7.5|3.2916666666666665|               2|             1|
|2015-06-19 00:00:00|45000.0|      14|          777|           525|           252|           5|252450|    5|2015-06-17 00:00:00|               767|                520|                247|                5|
       14|2015-06-22 00:00:00|               788|                531|                257|                5|            5|           -43|               10|                5|            5|           2|          11|
     6|              5|          3|    20.99|     17.99|                5.0|3.6666666666666665|               1|             1|
|2015-06-22 00:00:00|44000.0|     -17|          788|           531|           257|           5|252450|    6|2015-06-19 00:00:00|               777|                525|                252|                5|
      -17|2015-10-08 00:00:00|               884|                579|                305|                5|            5|            50|               11|                6|            5|           3|          96|
    48|             48|        108|    29.99|     20.99| 3.6666666666666665|0.8888888888888888|               2|             0|
|2015-10-16 00:00:00|48000.0|      50|          896|           583|           313|           5|252450|    8|2015-10-08 00:00:00|               884|                579|                305|                5|
       50|2015-11-20 00:00:00|               936|                609|                327|                5|            5|            66|               12|                4|            8|           8|          40|
    26|             14|         35|    29.99|     14.99|                1.5|1.1428571428571428|               1|             1|
|2015-11-22 00:00:00|52000.0|      66|          951|           617|           334|           5|252450|   10|2015-11-20 00:00:00|               936|                609|                327|                5|
       66|2015-11-25 00:00:00|               964|                624|                340|                5|            5|            70|               15|                8|            7|           2|          13|
     7|              6|          3|    29.99|     10.19|                7.5| 4.333333333333333|               1|             1|
+-------------------+-------+--------+-------------+--------------+--------------+------------+------+-----+-------------------+------------------+-------------------+-------------------+-----------------+---
---------+-------------------+------------------+-------------------+-------------------+-----------------+-------------+--------------+-----------------+-----------------+-------------+------------+---------
-----+---------------+-----------+---------+----------+-------------------+------------------+----------------+--------------+
only showing top 5 rows
```

14. Select features and rename columns

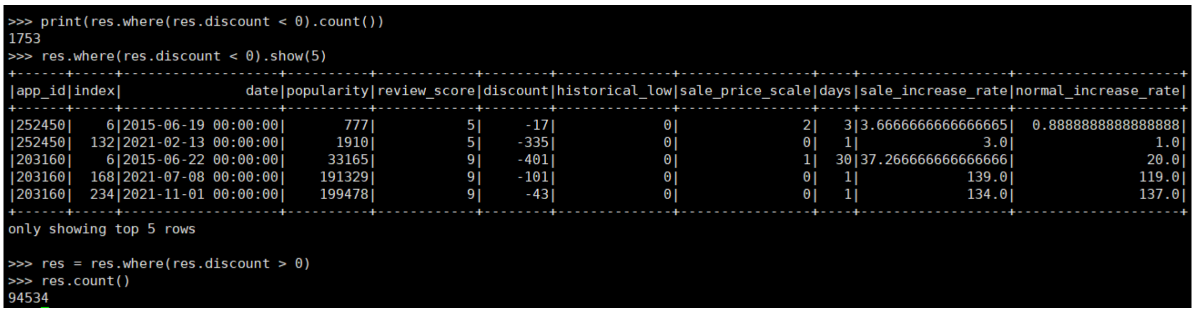
```
res = df.select('app_id', 'index', func.col('prev_date').alias('date'),
func.col('prev_total_reviews').alias('popularity'),
func.col('prev_review_score').alias('review_score'), 'discount',
'historical_low', 'sale_price_scale', func.col('days_increase').alias('days'),
func.col('total_increase_rate').alias('sale_increase_rate'),
func.col('total_normal_rate').alias('normal_increase_rate'))
res.show(5)
```



```
>>> res = df.select('app_id', 'index', func.col('prev_date').alias('date'), func.col('prev_total_reviews').alias('popularity'), func.col('prev_revi
, 'sale_price_scale', func.col('days_increase').alias('days'), func.col('total_increase_rate').alias('sale_increase_rate'), func.col('total_normal_
>>> res.show(5)
+------+-----+-------------------+----------+------------+--------+--------------+----------------+----+------------------+--------------------+
|app_id|index|               date|popularity|review_score|discount|historical_low|sale_price_scale|days|sale_increase_rate|normal_increase_rate|
+------+-----+-------------------+----------+------------+--------+--------------+----------------+----+------------------+--------------------+
|252450|    2|2015-05-14 00:00:00|       640|           6|      17|             1|               2|   4|               7.5|  3.2916666666666665|
|252450|    5|2015-06-17 00:00:00|       767|           5|      14|             1|               1|   2|               5.0|  3.6666666666666665|
|252450|    6|2015-06-19 00:00:00|       777|           5|     -17|             0|               2|   3|3.6666666666666665|  0.8888888888888888|
|252450|    8|2015-10-08 00:00:00|       884|           5|      50|             1|               1|   8|               1.5|  1.1428571428571428|
|252450|   10|2015-11-20 00:00:00|       936|           5|      66|             1|               1|   2|               7.5|   4.333333333333333|
+------+-----+-------------------+----------+------------+--------+--------------+----------------+----+------------------+--------------------+
only showing top 5 rows
```

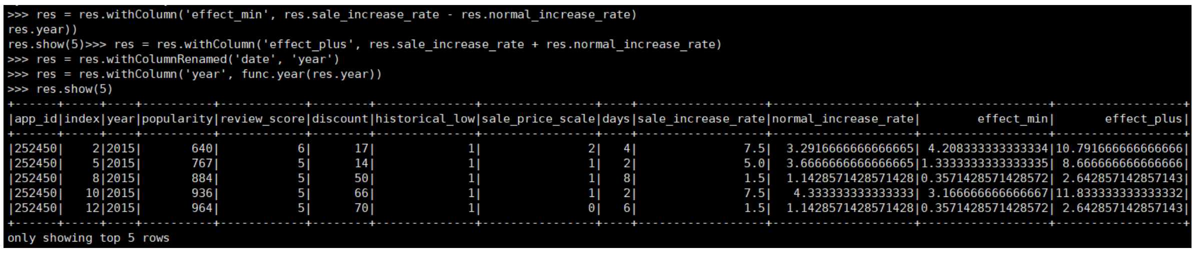
15. Clean discount < 0, error rows

```
print(res.where(res.discount < 0).count())
res.where(res.discount < 0).show(5)
res = res.where(res.discount > 0)
res.count()
```


```
>>> print(res.where(res.discount < 0).count())
1753
>>> res.where(res.discount < 0).show(5)
+------+-----+-------------------+----------+------------+--------+--------------+----------------+----+------------------+--------------------+
|app_id|index|               date|popularity|review_score|discount|historical_low|sale_price_scale|days|sale_increase_rate|normal_increase_rate|
+------+-----+-------------------+----------+------------+--------+--------------+----------------+----+------------------+--------------------+
|252450|    6|2015-06-19 00:00:00|       777|           5|     -17|             0|               2|   3|3.6666666666666665|  0.8888888888888888|
|252450|  132|2021-02-13 00:00:00|      1910|           5|    -335|             0|               0|   1|               3.0|                 1.0|
|203160|    6|2015-06-22 00:00:00|     33165|           9|    -401|             0|               1|  30|37.266666666666666|                20.0|
|203160|  168|2021-07-08 00:00:00|    191329|           9|    -101|             0|               0|   1|             139.0|               119.0|
|203160|  234|2021-11-01 00:00:00|    199478|           9|     -43|             0|               0|   1|             134.0|               137.0|
+------+-----+-------------------+----------+------------+--------+--------------+----------------+----+------------------+--------------------+
only showing top 5 rows

>>> res = res.where(res.discount > 0)
>>> res.count()
94534
```

16. Prepare Potential output features, cast date to year

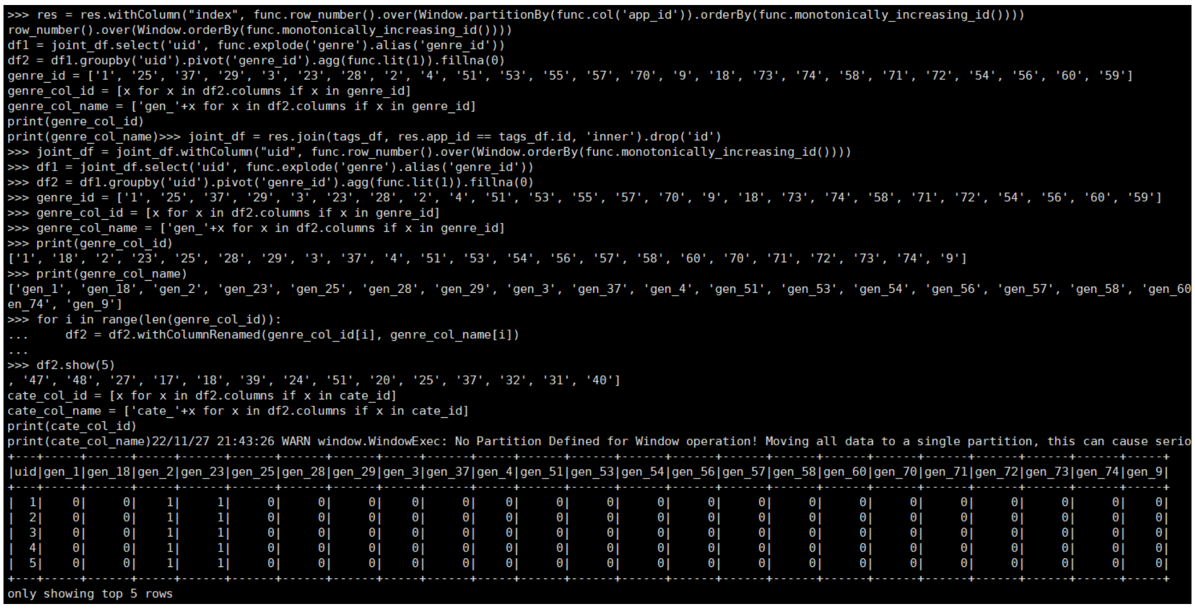
```
res = res.withColumn('effect_min', res.sale_increase_rate -
res.normal_increase_rate)
res = res.withColumn('effect_plus', res.sale_increase_rate +
res.normal_increase_rate)
res = res.withColumnRenamed('date', 'year')
res = res.withColumn('year', func.year(res.year))
res.show(5)
```


```
>>> res = res.withColumn('effect_min', res.sale_increase_rate - res.normal_increase_rate)
res.year))
res.show(5)>>> res = res.withColumn('effect_plus', res.sale_increase_rate + res.normal_increase_rate)
>>> res = res.withColumnRenamed('date', 'year')
>>> res = res.withColumn('year', func.year(res.year))
>>> res.show(5)
+------+-----+----+----------+------------+--------+--------------+----------------+----+------------------+--------------------+------------------+------------------+
|app_id|index|year|popularity|review_score|discount|historical_low|sale_price_scale|days|sale_increase_rate|normal_increase_rate|        effect_min|       effect_plus|
+------+-----+----+----------+------------+--------+--------------+----------------+----+------------------+--------------------+------------------+------------------+
|252450|    2|2015|       640|           6|      17|             1|               2|   4|               7.5|  3.2916666666666665| 4.208333333333334|10.791666666666666|
|252450|    5|2015|       767|           5|      14|             1|               1|   2|               5.0|  3.6666666666666665|1.3333333333333335| 8.666666666666666|
|252450|    8|2015|       884|           5|      50|             1|               1|   8|               1.5|  1.1428571428571428|0.3571428571428572| 2.642857142857143|
|252450|   10|2015|       936|           5|      66|             1|               1|   2|               7.5|   4.333333333333333| 3.166666666666667|11.833333333333332|
|252450|   12|2015|       964|           5|      70|             1|               0|   6|               1.5|  1.1428571428571428|0.3571428571428572| 2.642857142857143|
+------+-----+----+----------+------------+--------+--------------+----------------+----+------------------+--------------------+------------------+------------------+
only showing top 5 rows
```

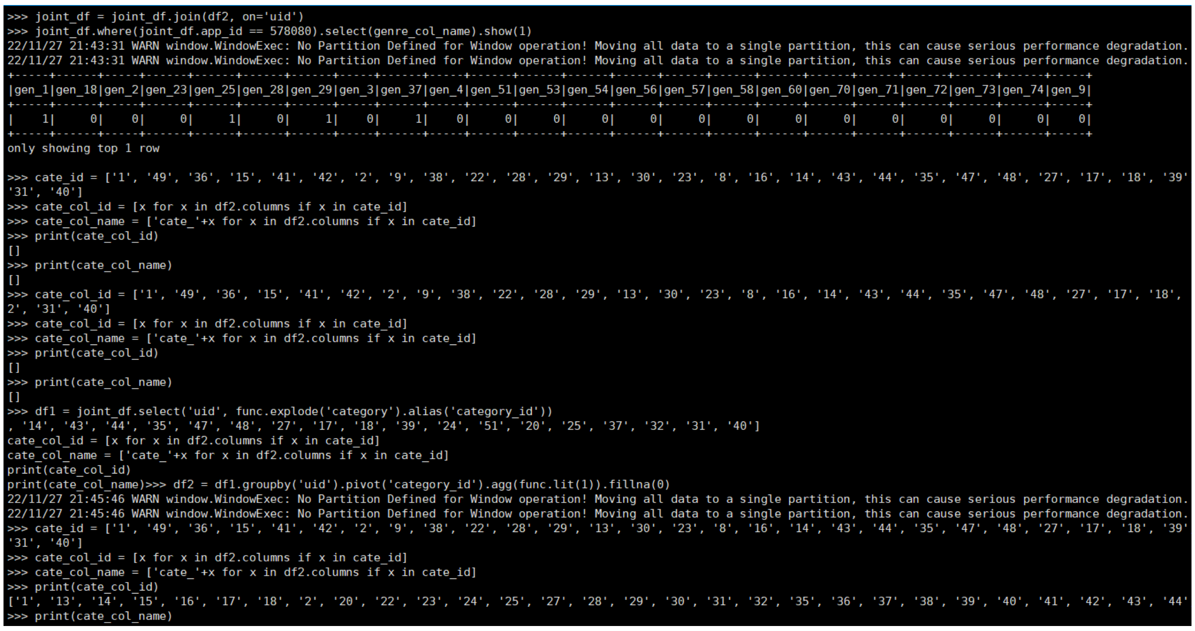
17. Get Genre and Category as Dummy Variables, Full Version of code in ETL single *.py Job

```
res = res.withColumn("index",
func.row_number().over(Window.partitionBy(func.col('app_id')).orderBy(func.monot
onically_increasing_id())))
joint_df = res.join(tags_df, res.app_id == tags_df.id, 'inner').drop('id')
joint_df = joint_df.withColumn("uid",
func.row_number().over(Window.orderBy(func.monotonically_increasing_id())))
df1 = joint_df.select('uid', func.explode('genre').alias('genre_id'))
df2 = df1.groupby('uid').pivot('genre_id').agg(func.lit(1)).fillna(0)
genre_id = ['1', '25', '37', '29', '3', '23', '28', '2', '4', '51', '53', '55',
'57', '70', '9', '18', '73', '74', '58', '71', '72', '54', '56', '60', '59']
genre_col_id = [x for x in df2.columns if x in genre_id]
genre_col_name = ['gen_'+x for x in df2.columns if x in genre_id]
print(genre_col_id)
print(genre_col_name)
for i in range(len(genre_col_id)):
    df2 = df2.withColumnRenamed(genre_col_id[i], genre_col_name[i])
df2.show(5)
joint_df = joint_df.join(df2, on='uid')
joint_df.where(joint_df.app_id == 578080).select(genre_col_name).show(1)

df1 = joint_df.select('uid', func.explode('category').alias('category_id'))
df2 = df1.groupby('uid').pivot('category_id').agg(func.lit(1)).fillna(0)
cate_id = ['1', '49', '36', '15', '41', '42', '2', '9', '38', '22', '28', '29',
'13', '30', '23', '8', '16', '14', '43', '44', '35', '47', '48', '27', '17',
'18', '39', '24', '51', '20', '25', '37', '32', '31', '40']
cate_col_id = [x for x in df2.columns if x in cate_id]
cate_col_name = ['cate_'+x for x in df2.columns if x in cate_id]
print(cate_col_id)
```

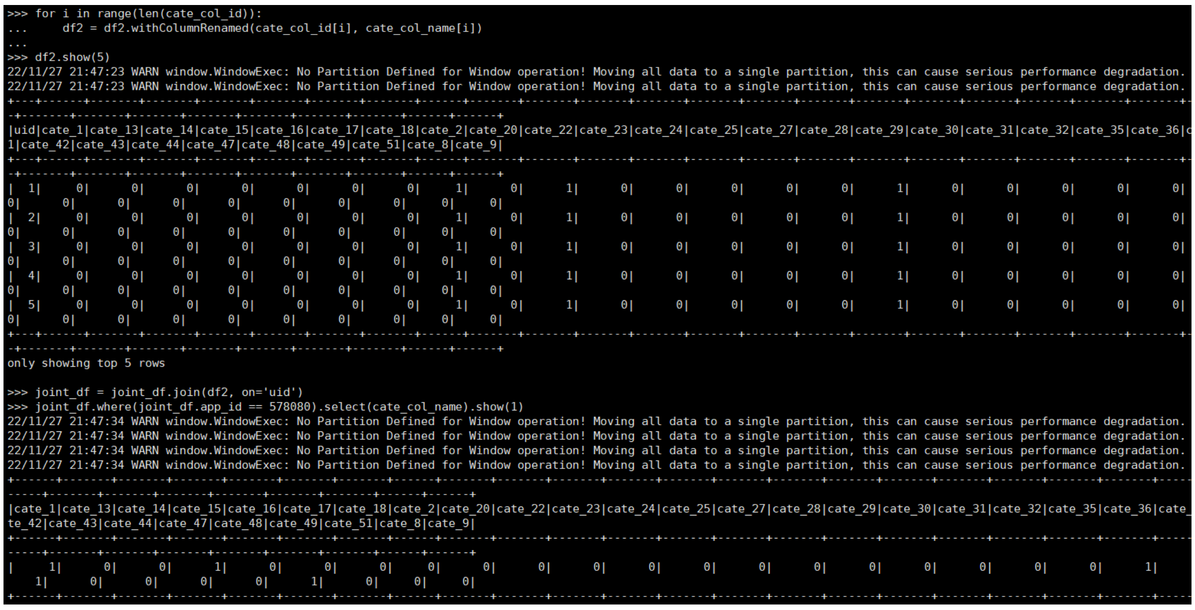
```python
print(cate_col_name)
for i in range(len(cate_col_id)):
    df2 = df2.withColumnRenamed(cate_col_id[i], cate_col_name[i])
df2.show(5)
joint_df = joint_df.join(df2, on='uid')
joint_df.where(joint_df.app_id == 578080).select(cate_col_name).show(1)
```



```
>>> res = res.withColumn("index", func.row_number().over(Window.partitionBy(func.col('app_id')).orderBy(func.monotonically_increasing_id())))
row_number().over(Window.orderBy(func.monotonically_increasing_id())))
df1 = joint_df.select('uid', func.explode('genre').alias('genre_id'))
df2 = df1.groupby('uid').pivot('genre_id').agg(func.lit(1)).fillna(0)
genre_id = ['1', '25', '37', '29', '3', '23', '28', '2', '4', '51', '53', '55', '57', '70', '9', '18', '73', '74', '58', '71', '72', '54', '56', '60', '59']
genre_col_id = [x for x in df2.columns if x in genre_id]
genre_col_name = ['gen_'+x for x in df2.columns if x in genre_id]
print(genre_col_id)
print(genre_col_name)>>> joint_df = res.join(tags_df, res.app_id == tags_df.id, 'inner').drop('id')
>>> joint_df = joint_df.withColumn("uid", func.row_number().over(Window.orderBy(func.monotonically_increasing_id())))
>>> df1 = joint_df.select('uid', func.explode('genre').alias('genre_id'))
>>> df2 = df1.groupby('uid').pivot('genre_id').agg(func.lit(1)).fillna(0)
>>> genre_id = ['1', '25', '37', '29', '3', '23', '28', '2', '4', '51', '53', '55', '57', '70', '9', '18', '73', '74', '58', '71', '72', '54', '56', '60', '59']
>>> genre_col_id = [x for x in df2.columns if x in genre_id]
>>> genre_col_name = ['gen_'+x for x in df2.columns if x in genre_id]
>>> print(genre_col_id)
['1', '18', '2', '23', '25', '28', '29', '3', '37', '4', '51', '53', '54', '56', '57', '58', '60', '70', '71', '72', '73', '74', '9']
>>> print(genre_col_name)
['gen_1', 'gen_18', 'gen_2', 'gen_23', 'gen_25', 'gen_28', 'gen_29', 'gen_3', 'gen_37', 'gen_4', 'gen_51', 'gen_53', 'gen_54', 'gen_56', 'gen_57', 'gen_58', 'gen_60
en_74', 'gen_9']
>>> for i in range(len(genre_col_id)):
...     df2 = df2.withColumnRenamed(genre_col_id[i], genre_col_name[i])
...
>>> df2.show(5)
, '47', '48', '27', '17', '18', '39', '24', '51', '20', '25', '37', '32', '31', '40']
cate_col_id = [x for x in df2.columns if x in cate_id]
cate_col_name = ['cate_'+x for x in df2.columns if x in cate_id]
print(cate_col_id)
print(cate_col_name)22/11/27 21:43:26 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serio
+---+-----+------+-----+------+------+------+------+-----+------+-----+------+------+------+------+------+------+------+------+------+------+------+------+-----+
|uid|gen_1|gen_18|gen_2|gen_23|gen_25|gen_28|gen_29|gen_3|gen_37|gen_4|gen_51|gen_53|gen_54|gen_56|gen_57|gen_58|gen_60|gen_70|gen_71|gen_72|gen_73|gen_74|gen_9|
+---+-----+------+-----+------+------+------+------+-----+------+-----+------+------+------+------+------+------+------+------+------+------+------+------+-----+
|  1|    0|     0|    1|     1|     0|     0|     0|    0|     0|    0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|    0|
|  2|    0|     0|    1|     1|     0|     0|     0|    0|     0|    0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|    0|
|  3|    0|     0|    1|     1|     0|     0|     0|    0|     0|    0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|    0|
|  4|    0|     0|    1|     1|     0|     0|     0|    0|     0|    0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|    0|
|  5|    0|     0|    1|     1|     0|     0|     0|    0|     0|    0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|    0|
+---+-----+------+-----+------+------+------+------+-----+------+-----+------+------+------+------+------+------+------+------+------+------+------+------+-----+
only showing top 5 rows
```



```
>>> joint_df = joint_df.join(df2, on='uid')
>>> joint_df.where(joint_df.app_id == 578080).select(genre_col_name).show(1)
22/11/27 21:43:31 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
22/11/27 21:43:31 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
+-----+------+-----+------+------+------+------+-----+------+-----+------+------+------+------+------+------+------+------+------+------+------+------+-----+
|gen_1|gen_18|gen_2|gen_23|gen_25|gen_28|gen_29|gen_3|gen_37|gen_4|gen_51|gen_53|gen_54|gen_56|gen_57|gen_58|gen_60|gen_70|gen_71|gen_72|gen_73|gen_74|gen_9|
+-----+------+-----+------+------+------+------+-----+------+-----+------+------+------+------+------+------+------+------+------+------+------+------+-----+
|    1|     0|    0|     0|     1|     0|     1|    0|     1|    0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|     0|    0|
+-----+------+-----+------+------+------+------+-----+------+-----+------+------+------+------+------+------+------+------+------+------+------+------+-----+
only showing top 1 row

>>> cate_id = ['1', '49', '36', '15', '41', '42', '2', '9', '38', '22', '28', '29', '13', '30', '23', '8', '16', '14', '43', '44', '35', '47', '48', '27', '17', '18', '39'
'31', '40']
>>> cate_col_id = [x for x in df2.columns if x in cate_id]
>>> cate_col_name = ['cate_'+x for x in df2.columns if x in cate_id]
>>> print(cate_col_id)
[]
>>> print(cate_col_name)
[]
>>> cate_col_id = ['1', '49', '36', '15', '41', '42', '2', '9', '38', '22', '28', '29', '13', '30', '23', '8', '16', '14', '43', '44', '35', '47', '48', '27', '17', '18',
2', '31', '40']
>>> cate_col_id = [x for x in df2.columns if x in cate_id]
>>> cate_col_name = ['cate_'+x for x in df2.columns if x in cate_id]
>>> print(cate_col_id)
[]
>>> print(cate_col_name)
[]
>>> df1 = joint_df.select('uid', func.explode('category').alias('category_id'))
, '14', '43', '44', '35', '47', '48', '27', '17', '18', '39', '24', '51', '20', '25', '37', '32', '31', '40']
cate_col_id = [x for x in df2.columns if x in cate_id]
cate_col_name = ['cate_'+x for x in df2.columns if x in cate_id]
print(cate_col_id)
print(cate_col_name)>>> df2 = df1.groupby('uid').pivot('category_id').agg(func.lit(1)).fillna(0)
22/11/27 21:45:46 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
22/11/27 21:45:46 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
>>> cate_id = ['1', '49', '36', '15', '41', '42', '2', '9', '38', '22', '28', '29', '13', '30', '23', '8', '16', '14', '43', '44', '35', '47', '48', '27', '17', '18', '39'
'31', '40']
>>> cate_col_id = [x for x in df2.columns if x in cate_id]
>>> cate_col_name = ['cate_'+x for x in df2.columns if x in cate_id]
>>> print(cate_col_id)
['1', '13', '14', '15', '16', '17', '18', '2', '20', '22', '23', '24', '25', '27', '28', '29', '30', '31', '32', '35', '36', '37', '38', '39', '40', '41', '42', '43', '44'
>>> print(cate_col_name)
```



```
>>> for i in range(len(cate_col_id)):
...     df2 = df2.withColumnRenamed(cate_col_id[i], cate_col_name[i])
...
>>> df2.show(5)
22/11/27 21:47:23 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
22/11/27 21:47:23 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
+---+------+-------+-------+-------+-------+-------+-------+------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-
-+-------+-------+-------+-------+-------+-------+-------+------+------+
|uid|cate_1|cate_13|cate_14|cate_15|cate_16|cate_17|cate_18|cate_2|cate_20|cate_22|cate_23|cate_24|cate_25|cate_27|cate_28|cate_29|cate_30|cate_31|cate_32|cate_35|cate_36|c
1|cate_42|cate_43|cate_44|cate_47|cate_48|cate_49|cate_51|cate_8|cate_9|
+---+------+-------+-------+-------+-------+-------+-------+------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-
-+-------+-------+-------+-------+-------+-------+-------+------+------+
|  1|     0|      0|      0|      0|      0|      0|      0|     1|      0|      1|      0|      0|      0|      0|      0|      1|      0|      0|      0|      0|      0|
0|      0|      0|      0|      0|      0|      0|      0|     0|     0|
|  2|     0|      0|      0|      0|      0|      0|      0|     1|      0|      1|      0|      0|      0|      0|      0|      1|      0|      0|      0|      0|      0|
0|      0|      0|      0|      0|      0|      0|      0|     0|     0|
|  3|     0|      0|      0|      0|      0|      0|      0|     1|      0|      1|      0|      0|      0|      0|      0|      1|      0|      0|      0|      0|      0|
0|      0|      0|      0|      0|      0|      0|      0|     0|     0|
|  4|     0|      0|      0|      0|      0|      0|      0|     1|      0|      1|      0|      0|      0|      0|      0|      1|      0|      0|      0|      0|      0|
0|      0|      0|      0|      0|      0|      0|      0|     0|     0|
|  5|     0|      0|      0|      0|      0|      0|      0|     1|      0|      1|      0|      0|      0|      0|      0|      1|      0|      0|      0|      0|      0|
0|      0|      0|      0|      0|      0|      0|      0|     0|     0|
+---+------+-------+-------+-------+-------+-------+-------+------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-
-+-------+-------+-------+-------+-------+-------+-------+------+------+
only showing top 5 rows

>>> joint_df = joint_df.join(df2, on='uid')
>>> joint_df.where(joint_df.app_id == 578080).select(cate_col_name).show(1)
22/11/27 21:47:34 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
22/11/27 21:47:34 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
22/11/27 21:47:34 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
22/11/27 21:47:34 WARN window.WindowExec: No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation.
+------+-------+-------+-------+-------+-------+-------+------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+----
-----+-------+-------+-------+-------+-------+-------+------+------+
|cate_1|cate_13|cate_14|cate_15|cate_16|cate_17|cate_18|cate_2|cate_20|cate_22|cate_23|cate_24|cate_25|cate_27|cate_28|cate_29|cate_30|cate_31|cate_32|cate_35|cate_36|cate_
te_42|cate_43|cate_44|cate_47|cate_48|cate_49|cate_51|cate_8|cate_9|
+------+-------+-------+-------+-------+-------+-------+------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+----
-----+-------+-------+-------+-------+-------+-------+------+------+
|     1|      0|      0|      1|      0|      0|      0|     0|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|      1|
    1|      0|      0|      0|      0|      1|      0|     0|     0|
+------+-------+-------+-------+-------+-------+-------+------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+----
```