flipkart-pyspark-project

November 23, 2024

[0]:

```python
# importing lib
from pyspark.sql import SparkSession
from pyspark.sql.functions import expr
from pyspark.sql.functions import col,lit,isnan,when,count
from pyspark.sql.functions import *
```

[0]:

```python
#Creating Spark Session
spark=SparkSession.builder.appName("Flipkart Data Engineering").getOrCreate()
```

[0]:

```python
#file path
file_path='/FileStore/tables/Flipkart-1.csv'

flipkart_df=spark.read.csv(file_path,header=True,inferSchema=True)
flipkart_df.display()
```

[0]:

```python
#Schema
flipkart_df.printSchema()
flipkart_df.describe().show()
```

```
root
 |-- id: integer (nullable = true)
 |-- title: string (nullable = true)
 |-- Rating: double (nullable = true)
 |-- maincateg: string (nullable = true)
 |-- platform: string (nullable = true)
 |-- actprice1: integer (nullable = true)
 |-- norating1: integer (nullable = true)
 |-- noreviews1: integer (nullable = true)
 |-- star_5f: integer (nullable = true)
 |-- star_4f: integer (nullable = true)
 |-- star_3f: integer (nullable = true)
 |-- star_2f: integer (nullable = true)
 |-- star_1f: integer (nullable = true)
 |-- fulfilled1: integer (nullable = true)

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

```
------------------+
|summary|                id|               title|
Rating|maincateg|platform|         actprice1|         norating1|
noreviews1|          star_5f|           star_4f|            star_3f|
star_2f|           star_1f|         fulfilled1|
+-------+------------------+--------------------+-------------------+---------+-
-------+------------------+------------------+------------------+---------------
--+------------------+------------------+------------------+------------------+--
------------------+
|  count|              5244|                5244|               5041|     5177|
5244|              5244|              5244|              5244|             5176|
5244|              5244|              5244|              5058|
5244|
|   mean|10507.372616323417|                 0.0|  4.011089069629038|     null|
null| 1378.657894736842|2988.5800915331806|415.49103737604884|1557.443199381762|
639.7854691075515| 356.3567887109077|154.13996948893973| 270.3977856860419|
0.6045003813882532|
| stddev|  5978.65889151765|                null|0.30191522284782074|     null|
null|1280.6300702165822|12881.253714820072|1910.7266693173326|6583.766997674775|
2991.065223081954|1632.7328338881507|
611.0067985620702|1035.0852878031521|0.48900436610958664|
|    min|                 0|"AADI MEN""S BLAC…|                0.0|      Men|
Amazon|               139|                 1|                 0|
0|                 0|                 0|                 0|                 0|
0|
|    max|             20964| Bellies For Wome…|                5.0|
Women|Flipkart|             15999|            289973|             45448|
151193|            74037|             49924|             12629|
23139|                  1|
+-------+------------------+--------------------+-------------------+---------+-
-------+------------------+------------------+------------------+---------------
--+------------------+------------------+------------------+------------------+--
------------------+
```

[0]:

```
#missing data

flipkart_df.select([count(when(col(c).isNull(), c)).alias(c) for c in␣
 ↪flipkart_df.columns]).display()
```

[0]:

```
#drop the rows that is missing
flipkart_df_clean=flipkart_df.dropna()


#filling specific values to the nan columns or missing columns
flipkart_df_filled=flipkart_df.fillna({"Rating":0,"maincateg":"Men"})
```

[0]:

```
# Filter products with ratings greater than 4 and priced below 1000
high_rated_products = flipkart_df_filled.filter((col("Rating") > 4) )

# Show the result
high_rated_products.display(5)
```

[0]:

```
#group by the category and calculte the average rating

avg_rating_by_category=flipkart_df_filled.groupBy("maincateg").avg("Rating")
avg_rating_by_category.display()
```

[0]:

```
#Total  Revenue by category

total_revenue_by_category=flipkart_df_filled.groupBy("maincateg").
  ↪agg(sum("Rating"))
total_revenue_by_category.display()
```

[0]:

```
#Save the Processed Data

output_table='Flipkart_Data_Analysis_table'
flipkart_df_filled.write.mode("overwrite").saveAsTable(output_table)
```

[0]:

```
%sql
select * from flipkart_data_analysis_table limit 10
```

[0]:

```

```