# Importing Libraries

```
!pip install pyspark
from pyspark.sql import SparkSession
from pyspark.sql.types import *
import pandas as pd
import matplotlib.pyplot as plt
import·seaborn·as·sns
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Collecting pyspark
  Downloading pyspark-3.3.2.tar.gz (281.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 281.4/281.4 MB 5.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting py4j==0.10.9.5
  Downloading py4j-0.10.9.5-py2.py3-none-any.whl (199 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 199.7/199.7 kB 13.7 MB/s eta 0:00:00
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.3.2-py2.py3-none-any.whl size=281824
  Stored in directory: /root/.cache/pip/wheels/6c/e3/9b/0525ce8a69478916513509d43693
Successfully built pyspark
Installing collected packages: py4j, pyspark
  Attempting uninstall: py4j
    Found existing installation: py4j 0.10.9.7
    Uninstalling py4j-0.10.9.7:
      Successfully uninstalled py4j-0.10.9.7
Successfully installed py4j-0.10.9.5 pyspark-3.3.2
```

```
spark = SparkSession.builder.appName('AmazonReviews').getOrCreate()
```

```
import pyspark
from pyspark.context import SparkContext

from pyspark import SparkConf
sc = SparkContext.getOrCreate(SparkConf().setMaster("local[4]"))
```

# Connecting The Collab With Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
import sklearn
```

## Loading The Dataset

```
df = spark.read.csv('/content/drive/MyDrive/Amazon_Unlocked_Mobile.csv', header=True, infe
```

## Displaying 1st 5 Data

```
# display the dataset
df.show(5)
```

```
+--------------------+----------+------+------+--------------------+------------+
|        Product Name|Brand Name| Price|Rating|             Reviews|Review Votes|
+--------------------+----------+------+------+--------------------+------------+
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|I feel so LUCKY t...|           1|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|nice phone, nice ...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|        Very pleased|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|It works good but...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|Great phone to re...|           0|
+--------------------+----------+------+------+--------------------+------------+
only showing top 5 rows
```

## Checking Null Values

```
from pyspark.sql.functions import isnan, when, count, col

# Count the number of null values in each column

df.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in df.columns]).show(
```

```
+------------+----------+-----+------+-------+------------+
|Product Name|Brand Name|Price|Rating|Reviews|Review Votes|
+------------+----------+-----+------+-------+------------+
|           0|     64376|    6|   379|     78|       11531|
+------------+----------+-----+------+-------+------------+
```

we have 0 null values in product name column

we have 64376 null values in brand name column we have 6 null values in price column

we have 379 null values in rating column

we have 78 null values in reviews column

we have 11531 null values in review votes column

## Replacing Nulll Value

```
# Fill null values with a default value

df = df.fillna({'Price': 0.0, 'Brand Name': 'Unknown', 'Reviews': 0, 'Rating': 0, 'Review

# Verify that there are no more null values

df.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in df.columns]).show(
```

```
+------------+----------+-----+------+-------+------------+
|Product Name|Brand Name|Price|Rating|Reviews|Review Votes|
+------------+----------+-----+------+-------+------------+
|           0|         0|    0|     0|      0|           0|
+------------+----------+-----+------+-------+------------+
```

we have replaced the null value with default value in each column

as we have 0 null value in product name column we are not going to do anything

we replaced the null values in Price column with 0.0

we replaced the null values in Brand Name column with Unknown

we replaced the null values in Reviews column with 0

we replaced the null value in Rating column with 0

we relaced the null value in Review Votes column with 0

## NO OF RECORDS

```
# total value

df.count()
```

```
413848
```

we have 4,13,848 records

## NO OF COLUMNS

```
# total column

len(df.columns)

    6
```

we have 6 columns

## COLUMN NAMES

```
# column name

df.columns

    ['Product Name', 'Brand Name', 'Price', 'Rating', 'Reviews', 'Review Votes']
```

```
# no of record grouped by product name

df.groupBy("Product Name").count().show()

    +--------------------+-----+
    |        Product Name|count|
    +--------------------+-----+
    |Apple iPhone 4 A1...|  330|
    |Apple iPhone 6s 1...|  163|
    |"BLU Studio M HD ...|   47|
    |BlueCosmo Iridium...|    1|
    |"Cellphones Unloc...|   40|
    |CNPGD® All-in-1 S...|  261|
    |Flip Phone Unlock...|    2|
    |H2O Nano SIM Card...|    1|
    |LG H955 Unlocked ...|   44|
    |LG LS670 OPTIMUS ...|    1|
    |"LG Nexus 5 D820 ...|   62|
    |4G-Unlocked Huawe...|    7|
    |Apple iPhone 5C Y...|   10|
    |Apple iPhone 5s U...|   10|
    |Apple iPhone 6 Pl...|  176|
    |Apple iPhone SE 6...|   12|
    |ASUS ZenFone 2 Un...|   10|
    |Blackberry 9530 S...|  484|
    |BlackBerry Torch ...|   14|
    |BLU Win JR Smartp...|   42|
    +--------------------+-----+
    only showing top 20 rows
```

330 people has bought apple iphone

1 person has bought LG LS670 OPTIMUS

```
# no of record grouped by brand name

df.groupBy("Brand Name").count().show()
```

```
    +--------------------+-----+
    |          Brand Name|count|
    +--------------------+-----+
    |              DOOGEE|   97|
    |                 H2O|    1|
    |             Getnord|    5|
    |                Kata|   40|
    |                P710|   30|
    | Android 4.1 - In...|   36|
    |               Nokia|16086|
    |                4GB"|    1|
    |             LandRum|    6|
    |             Ulefone|  141|
    |            px phone|   84|
    |               JIAKE|   32|
    |                13MP|  123|
    |          Jelly Bean|   42|
    |8gb - Internation...|  138|
    |         AeroAntenna|    1|
    |      Android 4.4 KK| 1390|
    |                Doro|   21|
    |             Maxwest|   15|
    |                 htc|  203|
    +--------------------+-----+
    only showing top 20 rows
```

16086 people has bought nokia brand

1 person has bought AeroAntenna , H2O brand

```
# phone with the price >= 2000

price_lesser_than_1000= df.filter(df['Price'] > 2000).show()
```

```
    +--------------------+----------+-----+------+--------------------+------------+
    |        Product Name|Brand Name|Price|Rating|             Reviews|Review Votes|
    +--------------------+----------+-----+------+--------------------+------------+
    |BlueCosmo Iridium...|   Iridium| 2598|     5|These folks are g...|           0|
    |"Huawei Ascend P7...|    Huawei| 2066|     1|This phonesoftwar...|           0|
|"Huawei Ascend P7...|    Huawei| 2066|     5|       great product|           0|
|"Huawei Ascend P7...|    Huawei| 2066|     5|All very good, ex...|           0|
|"Huawei Ascend P7...|    Huawei| 2066|     5|Super productTwo ...|           0|
|"Huawei Ascend P7...|    Huawei| 2066|     1|item were not as ...|           3|
|"Huawei Ascend P7...|    Huawei| 2066|     5|Great phone, grea...|           1|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     3|good phone. I am ...|           0|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     4|Very good product...|           0|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     1|the phone does no...|           0|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     5|  excelente producto|           0|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     1|The worst phone e...|           0|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     5|            excelent|           0|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     5|           Very good|           0|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     5|          Excelente¡|           0|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     5|                Good|           0|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     5|              i like|           0|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     5|lenovo recommend ...|           0|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     5|           Excelente|           0|
|Lenovo S8 S898T 5...|    Lenovo| 2224|     2|         Do not work|           0|
+--------------------+----------+-----+------+--------------------+------------+
only showing top 20 rows
```

maximum every phone has rating 5

so it's best to buy phone at the rate greater than 2000 price

```
# phone with highest and lowest price

from pyspark.sql.functions import min , max

df.select(max('Price'),min('Price')).show()
```

```
+----------+----------------+
|max(Price)|      min(Price)|
+----------+----------------+
|  verykool| 1 GHZ Dual Core |
+----------+----------------+
```

verykool mobile phone has the highest price

1 GHZ Dual Core mobile phone has the lowest price

```
# phone with 5 rating

rat_five= df.filter(df['Rating'] == 5).show()
```

```
+--------------------+----------+------+------+--------------------+------------+
|        Product Name|Brand Name| Price|Rating|             Reviews|Review Votes|
+--------------------+----------+------+------+--------------------+------------+
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|I feel so LUCKY t...|           1|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|        Very pleased|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|I originally was ...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|This is a great p...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|These guys are th...|           2|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|Ordered this phon...|           1|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|I was able to get...|           6|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|I brought this ph...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|the phone was gre...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|Phone works great...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|as described, fas...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|Perfect in every ...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|Just got this pho...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|The phone was gre...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|This phone came i...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|Met all of my exp...|           0|
|((Unlocked))Black...|   Unknown| 269.1|     5|Great. Arrived qu...|           0|
|((Unlocked))Black...|   Unknown| 269.1|     5|Avianna LLC is an...|           0|
|((Unlocked))Black...|   Unknown| 269.1|     5|Exactly what I wa...|           1|
|((Unlocked))Black...|   Unknown| 269.1|     5|Got it faster tha...|           0|
+--------------------+----------+------+------+--------------------+------------+
only showing top 20 rows
```

samsung product has the highest rating - 5

so it's best to buy samsung product

```
# phone with 4 rating

rat_five= df.filter(df['Rating'] == 4).show()
```

```
+--------------------+----------+------+------+--------------------+------------+
|        Product Name|Brand Name| Price|Rating|             Reviews|Review Votes|
+--------------------+----------+------+------+--------------------+------------+
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|nice phone, nice ...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|It works good but...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|Great phone to re...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|I love the phone....|           1|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|The battery was o...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|pros-beautiful sc...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|Phone good just a...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|Phone's speaker l...|           0|
|((Unlocked))Black...|   Unknown| 269.1|     4|             I liked|           0|
|((Unlocked))Black...|   Unknown| 269.1|     4|Phone works great...|           0|
|"(LANDVO) 5.0"" C...|       HTM| 69.99|     4|All around good p...|           0|
|"(LANDVO) 5.0"" C...|       HTM| 69.99|     4|I have no problem...|           0|
|"(LANDVO) 5.0"" C...|       HTM| 69.99|     4|           MUY BUENO|           0|
|"(LANDVO) 5.0"" C...|       HTM| 69.99|     4|Its great for the...|           1|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     4|Nice phone. Easy ...|           6|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     4|This is a great, ...|           9|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     4|easy to use. My m...|           1|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     4|"I bought it for ...|    I assume|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     4|simple to use. Do...|           0|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     4|Good phone for my...|           1|
+--------------------+----------+------+------+--------------------+------------+
only showing top 20 rows
```

```
# phone with 3 rating

rat_five= df.filter(df['Rating'] == 3).show()
```

```
+--------------------+----------+------+------+--------------------+----------------
|        Product Name|Brand Name| Price|Rating|             Reviews|        Review V
+--------------------+----------+------+------+--------------------+----------------
|"""CLEAR CLEAN ES...|   Samsung|199.99|     3|It's battery life...|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     3|My fiance had thi...|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     3|unfortunately Spr...|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     3|the reasons for t...|
|((Unlocked))Black...|   Unknown| 269.1|     3|Ad advertised as ...|
|"(LANDVO) 5.0"" C...|       HTM| 69.99|     3|Valid for Movilne...|
|"(LANDVO) 5.0"" C...|       HTM| 69.99|     3|The phone works g...|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     3|the charger did n...|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     3|         No internet|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     3|The only reason I...|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     3|good phone for my...|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     3|Not as sensitive ...|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     3|This may be an is...|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     3|"Word to the wise...| like Sprint. It
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     3|Phone number come...|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     3|Bought this for m...|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     3|returned would no...|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     3|My 79 year old mo...|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     3|It worked alright...|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     3|"Word to the wise...| like Sprint. It
+--------------------+----------+------+------+--------------------+----------------
only showing top 20 rows
```

```
# phone with 2 rating

rat_five= df.filter(df['Rating'] == 2).show()
```

```
+--------------------+----------+------+------+--------------------+------------+
|        Product Name|Brand Name| Price|Rating|             Reviews|Review Votes|
+--------------------+----------+------+------+--------------------+------------+
|"""CLEAR CLEAN ES...|   Samsung|199.99|     2|The charging port...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     2|Phone looks good ...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     2|Had this phone be...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     2|One of the phones...|           0|
|((Unlocked))Black...|   Unknown| 269.1|     2|when i got phone ...|           0|
|"(LANDVO) 5.0"" C...|       HTM| 69.99|     2|Delivery was fast...|           0|
|"(LANDVO) 5.0"" C...|       HTM| 69.99|     2|When I first got ...|           0|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     2|I like the FM rad...|           0|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     2|does not work wel...|           1|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     2|Sounds like you a...|           0|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     2|the phone is unab...|           1|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     2|sound quality poo...|           0|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     2|The speakers are ...|           0|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     2|My Granny couldn'...|           1|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     2|Not good sound, n...|           0|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     2|The SIM card from...|           2|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     2|I bought this pho...|           0|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     2|We bought the pho...|           0|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     2|As arrival of thi...|           1|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     2|Pros:* The cradle...|           0|
+--------------------+----------+------+------+--------------------+------------+
only showing top 20 rows
```

```
# phone with 1 rating

rat_five= df.filter(df['Rating'] == 1).show()
```

```
+--------------------+----------+------+------+--------------------+------------+
|        Product Name|Brand Name| Price|Rating|             Reviews|Review Votes|
+--------------------+----------+------+------+--------------------+------------+
|"""CLEAR CLEAN ES...|   Samsung|199.99|     1|I already had a p...|           1|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     1|I'm really disapp...|           1|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     1|I purchased this ...|          19|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     1|was not in good c...|           0|
|"""CLEAR CLEAN ES...|   Samsung|199.99|     1|Just... not good....|           0|
|"(LANDVO) 5.0"" C...|       HTM| 69.99|     1|Worked OK for awh...|           0|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     1|More complicated ...|           2|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     1|phone reception p...|           1|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     1|I was contacting ...|           1|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     1|Bought this phone...|           1|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     1|I searched for un...|           1|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     1|I am very unhappy...|           0|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     1|Shortly after ret...|           0|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     1|Stopped working a...|           0|
|"[XMAS DEAL] [New...|    Jethro| 79.99|     1|Defective product...|           2|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     1|I bought this pho...|           0|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     1|Ordered phone but...|           0|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     1|Returning product...|           0|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     1|Locks accidentall...|           0|
|[XMAS DEAL] Jethr...|    Jethro| 59.99|     1|Very bad same wee...|           0|
+--------------------+----------+------+------+--------------------+------------+
only showing top 20 rows
```

```
# no of record grouped by reviews

df.groupBy("Reviews").count().show()
```

```
+--------------------+-----+
|             Reviews|count|
+--------------------+-----+
|It was really bad...|    2|
|I took this phone...|    1|
|On point and fair...|    2|
|After only three ...|    1|
|Good phone..i jus...|    1|
|I bought the phon...|    3|
|Great condition a...|    4|
|the iphone 3gs is...|    3|
|Exactly as I expe...|    2|
|After having a fe...|    1|
|seller was great ...|    1|
|Do NOT get this p...|    1|
|The phone is very...|    1|
|Got to my house e...|    1|
|did not function ...|    3|
|I am happy with m...|    2|
|Having a great ti...|    2|
|My iphone 4 arriv...|    2|
|good items, the c...|    2|
|It's an iPhone, s...|    1|
+--------------------+-----+
only showing top 20 rows
```

```
from pyspark.sql.functions import when
from pyspark.sql.functions import lit
```

```
df.withColumn("sentiment", \
   when((df.Rating > 3), lit("positive")) \
     .when((df.Rating < 3), lit("negative")) \
     .otherwise(lit("neutral")) \
  ).show()
```

```
+--------------------+----------+------+------+--------------------+------------+---
|        Product Name|Brand Name| Price|Rating|             Reviews|Review Votes|sen
+--------------------+----------+------+------+--------------------+------------+---
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|I feel so LUCKY t...|           1| po
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|nice phone, nice ...|           0| po
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|        Very pleased|           0| po
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|It works good but...|           0| po
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|Great phone to re...|           0| po
|"""CLEAR CLEAN ES...|   Samsung|199.99|     1|I already had a p...|           1| ne
|"""CLEAR CLEAN ES...|   Samsung|199.99|     2|The charging port...|           0| ne
|"""CLEAR CLEAN ES...|   Samsung|199.99|     2|Phone looks good ...|           0| ne
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|I originally was ...|           0| po
|"""CLEAR CLEAN ES...|   Samsung|199.99|     3|It's battery life...|           0|  n
|"""CLEAR CLEAN ES...|   Samsung|199.99|     3|My fiance had thi...|           0|  n
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|This is a great p...|           0| po
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|These guys are th...|           2| po
|"""CLEAR CLEAN ES...|   Samsung|199.99|     1|I'm really disapp...|           1| ne
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|Ordered this phon...|           1| po
|"""CLEAR CLEAN ES...|   Samsung|199.99|     2|Had this phone be...|           0| ne
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|I was able to get...|           6| po
|"""CLEAR CLEAN ES...|   Samsung|199.99|     5|I brought this ph...|           0| po
|"""CLEAR CLEAN ES...|   Samsung|199.99|     4|I love the phone....|           1| po
|"""CLEAR CLEAN ES...|   Samsung|199.99|     3|unfortunately Spr...|           0|  n
+--------------------+----------+------+------+--------------------+------------+---
only showing top 20 rows
```

```
df.filter(df['Reviews'] == "great phone").show()
```

```
+--------------------+----------+------+------+-----------+------------+
|        Product Name|Brand Name| Price|Rating|    Reviews|Review Votes|
+--------------------+----------+------+------+-----------+------------+
|Apple Iphone 4 - ...|   Unknown|    NA|     5|great phone|           0|
|Apple iPhone 4 16...|     Apple|208.79|     5|great phone|           0|
|Apple iPhone 4 16...|     Apple|208.79|     5|great phone|           0|
|Apple iPhone 4 32...|     Apple| 99.99|     5|great phone|           0|
|Apple iPhone 4S 3...|     Apple|209.48|     5|great phone|           0|
|Apple iPhone 4S 6...|     Apple|   114|     5|great phone|           0|
|Apple iPhone 4S 6...|     Apple|   114|     5|great phone|           0|
|Apple iPhone 4s a...|   Unknown|159.99|     5|great phone|           0|
|Apple iPhone 5 Un...|     Apple|   265|     5|great phone|           0|
|Apple iPhone 5 Un...|     Apple|   309|     5|great phone|           0|
|Apple iPhone 5 Un...|     Apple|   309|     5|great phone|           0|
|Apple iPhone 5 Un...|     Apple|314.95|     5|great phone|           0|
|Apple iPhone 5C 1...|   Unknown|149.99|     5|great phone|           0|
|Apple iPhone 5s 1...|     Apple|149.99|     5|great phone|           0|
|Apple iPhone 5s 3...|     Apple|   125|     5|great phone|           0|
|Apple iPhone 5s 3...|     Apple|   209|     5|great phone|           0|
|Apple iPhone 5s 3...|     Apple|   209|     5|great phone|           0|
|Apple iPhone 5s 3...|     Apple|    49|     5|great phone|           0|
|Apple iPhone 5s 6...|     Apple|239.95|     5|great phone|           0|
|Apple iPhone 5s F...|     Apple|272.99|     5|great phone|           0|
+--------------------+----------+------+------+-----------+------------+
only showing top 20 rows
```

```
df.filter(df['Reviews'] == "not good").show()
```

```
+--------------------+----------+------+------+--------+------------+
|        Product Name|Brand Name| Price|Rating| Reviews|Review Votes|
+--------------------+----------+------+------+--------+------------+
|4 Inch Touch Scre...|   Unknown|  23.9|     1|not good|           0|
|4 Inch Touch Scre...|   Unknown|  23.9|     1|not good|           0|
|Apple iPhone 5 Un...|     Apple|   309|     1|not good|           0|
|Apple iPhone 5 Un...|     Apple|314.95|     1|not good|           0|
|Apple iPhone 5 Un...|     Apple|314.95|     1|not good|           0|
|Apple iPhone 6 Pl...|   Unknown|699.95|     2|not good|           1|
|Apple iPhone 6 Pl...|     Apple|   615|     2|not good|           1|
|Apple iPhone 6 Pl...|     Apple|   605|     2|not good|           1|
|Apple iPhone 6 Pl...|     Apple|   519|     2|not good|           1|
|Apple iPhone 6 Pl...|   Unknown|   490|     2|not good|           1|
|BLU Studio 5.0 HD...|       BLU|107.98|     1|not good|           0|
|BLU Studio 5.0 HD...|       BLU|119.99|     1|not good|           0|
|LG Neon GT365 Pre...|        LG| 69.99|     1|not good|           0|
|Motorola Droid RA...|  Motorola| 68.34|     1|not good|           0|
|Nokia 6350 Gray A...|     Nokia| 269.1|     1|not good|           0|
|Nokia C2-01.5 Unl...|     Nokia|    98|     1|not good|           0|
|"Samsung Galaxy G...|   Samsung|434.99|     2|not good|           0|
|Samsung Galaxy S6...|   Samsung|   529|     2|not good|           0|
|Samsung Galaxy S6...|   Samsung|   529|     2|not good|           0|
|Samsung Galaxy S6...|   Samsung|   449|     2|not good|           0|
+--------------------+----------+------+------+--------+------------+
     only showing top 20 rows
```

```
from pyspark.ml.feature import Tokenizer, StopWordsRemover, HashingTF, IDF, StringIndexer
from pyspark.ml.classification import NaiveBayes, LogisticRegression, RandomForestClassifi
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml.pipeline import Pipeline
from pyspark.mllib.evaluation import MulticlassMetrics
```

```
#Naive Bayes classifier
tokenizer = Tokenizer(inputCol='Review Votes', outputCol='words')
remover = StopWordsRemover(inputCol=tokenizer.getOutputCol(), outputCol='filtered')
hashingTF = HashingTF(inputCol=remover.getOutputCol(), outputCol='rawFeatures', numFeature
idf = IDF(inputCol=hashingTF.getOutputCol(), outputCol='features')
labelIndexer = StringIndexer(inputCol='Rating', outputCol='label', handleInvalid='keep')

nb = NaiveBayes()

pipeline = Pipeline(stages=[tokenizer, remover, hashingTF, idf, labelIndexer, nb])

(trainingData, testData) = df.randomSplit([0.7, 0.3], seed=123)

model = pipeline.fit(trainingData)

predictions = model.transform(testData)

evaluator = MulticlassClassificationEvaluator(labelCol='label', predictionCol='prediction'
accuracy = evaluator.evaluate(predictions)

print(f'Accuracy: {accuracy}')
```

```
     Accuracy: 0.5200615047738653
```

```
# Calculate evaluation metrics
metrics = MulticlassMetrics(predictions.select('prediction', 'label').rdd)
prec = metrics.weightedPrecision
rec = metrics.weightedRecall
f1 = metrics.weightedFMeasure()
```

```
# Print evaluation metrics
print(f'Precision: {prec}')
print(f'Recall: {rec}')
print(f'F1 Score: {f1}')
```

```
     Precision: 0.4379434522510633
     Recall: 0.5200615047738654
     F1 Score: 0.39322363287054907
```

```
#Logistic regression classifier
lr = LogisticRegression(maxIter=10, regParam=0.01)

pipeline = Pipeline(stages=[tokenizer, remover, hashingTF, idf, labelIndexer, lr])

(trainingData, testData) = df.randomSplit([0.7, 0.3], seed=123)

model = pipeline.fit(trainingData)

predictions = model.transform(testData)

evaluator = MulticlassClassificationEvaluator(labelCol='label', predictionCol='prediction'
accuracy = evaluator.evaluate(predictions)

print(f'Accuracy: {accuracy}')

# Calculate evaluation metrics
metrics = MulticlassMetrics(predictions.select('prediction', 'label').rdd)
prec = metrics.weightedPrecision
rec = metrics.weightedRecall
f1 = metrics.weightedFMeasure()


# Print evaluation metrics
print(f'Precision: {prec}')
print(f'Recall: {rec}')
print(f'F1 Score: {f1}')
```

```
Accuracy: 0.5396560884895909
Precision: 0.48746599312436023
Recall: 0.5396560884895908
F1 Score: 0.3901890112525952
```

```
#Random forest classifier
rf = RandomForestClassifier(numTrees=10)

pipeline = Pipeline(stages=[tokenizer, remover, hashingTF, idf, labelIndexer, rf])

(trainingData, testData) = df.randomSplit([0.7, 0.3], seed=123)

model = pipeline.fit(trainingData)

predictions = model.transform(testData)

evaluator = MulticlassClassificationEvaluator(labelCol='label', predictionCol='prediction'
accuracy = evaluator.evaluate(predictions)

print(f'Random forest classifier Accuracy: {accuracy}')

# Calculate evaluation metrics
```

```
metrics = MulticlassMetrics(predictions.select('prediction', 'label').rdd)
prec = metrics.weightedPrecision
rec = metrics.weightedRecall
f1 = metrics.weightedFMeasure()


# Print evaluation metrics
print(f'Precision: {prec}')
print(f'Recall: {rec}')
print(f'F1 Score: {f1}')
```

```
Random forest classifier Accuracy: 0.5223961100645639
Precision: 0.4173603011272721
Recall: 0.5223961100645639
F1 Score: 0.3586813433651473
```

```
#Decision Tree classifier
from pyspark.ml.classification import DecisionTreeClassifier

dt = DecisionTreeClassifier(labelCol='label', featuresCol='features')

pipeline = Pipeline(stages=[tokenizer, remover, hashingTF, idf, labelIndexer, dt])

(trainingData, testData) = df.randomSplit([0.7, 0.3], seed=123)

model = pipeline.fit(trainingData)

predictions = model.transform(testData)

evaluator = MulticlassClassificationEvaluator(labelCol='label', predictionCol='prediction'
accuracy = evaluator.evaluate(predictions)

print(f'Accuracy: {accuracy}')

# Calculate evaluation metrics
metrics = MulticlassMetrics(predictions.select('prediction', 'label').rdd)
prec = metrics.weightedPrecision
rec = metrics.weightedRecall
f1 = metrics.weightedFMeasure()


# Print evaluation metrics
print(f'Precision: {prec}')
print(f'Recall: {rec}')
print(f'F1 Score: {f1}')
```

```
Accuracy: 0.5290698610507334
Precision: 0.3314578038398621
Recall: 0.5290698610507334
F1 Score: 0.3674930330395204
```

```
# K-means clustering
from pyspark.ml.clustering import KMeans

kmeans = KMeans().setK(2).setSeed(1)
pipeline = Pipeline(stages=[tokenizer, remover, hashingTF, idf, labelIndexer, kmeans])

(trainingData, testData) = df.randomSplit([0.7, 0.3], seed=123)

model = pipeline.fit(trainingData)

predictions = model.transform(testData).withColumn("prediction", col("prediction").cast("d

evaluator = MulticlassClassificationEvaluator(labelCol='label', predictionCol='prediction'
accuracy = evaluator.evaluate(predictions)

print(f'Accuracy: {accuracy}')

# Calculate evaluation metrics
metrics = MulticlassMetrics(predictions.select('prediction', 'label').rdd)
prec = metrics.weightedPrecision
rec = metrics.weightedRecall
f1 = metrics.weightedFMeasure()


# Print evaluation metrics
print(f'Precision: {prec}')
print(f'Recall: {rec}')
print(f'F1 Score: {f1}')
```

```
Accuracy: 0.5218889371910673
Precision: 0.27237244815661843
Recall: 0.5218889371910673
F1 Score: 0.35793800406490034
```