Big Data Hadoop and Spark Developer Project

Project 3: Market Analysis in Banking Domain

Submitted by: Gairik Chakraborty

Problem Statement:

Your client, a Portuguese banking institution, ran a marketing campaign to convince potential customers to invest in a bank term deposit scheme.

The marketing campaigns were based on phone calls. Often, the same customer was contacted more than once through phone, in order to assess if they would want to subscribe to the bank term deposit or not. You have to perform the marketing analysis of the data generated by this campaign.

Domain: Banking (Market Analysis)

Loading the Dataset and Cleaning the data:

- Open FTP from Practice lab and create a folder named "**Project_data**" in the root directory where the project data is to be uploaded. Upload the .csv file in the **/Project_data** directory.
- Open Web Console from Practice lab and trace the file, change the dataset to a .**txt** format from a **.csv** file. Put the dataset (**project_3_data.txt**) in HDFS in **project/** directory.
- Create a rdd **temprdd** to store the **project_3_data.txt** dataset and on top of it Map HOF is executed to remove the double quotes from every element and the clean data is stored back in HDFS for further use.

```
ip-10-0-42-218 login: gairikchakraborty16gmail
Password:
Last login: Fri Apr  8 22:42:30 on pts/674
[gairikchakraborty16gmail@ip-10-0-42-218 ~]$ ls
emp576ok.java  employee.txt  pig_1647322712035.log  pig_1647329689840.log  pig_1647332941645.log  tmp
employee.java  Gairik        pig_1647323283454.log  pig_1647330782105.log  Project_data
[gairikchakraborty16gmail@ip-10-0-42-218 ~]$ cd Project_data
[gairikchakraborty16gmail@ip-10-0-42-218 Project_data]$ ls
Project_3_data.csv
[gairikchakraborty16gmail@ip-10-0-42-218 Project_data]$ mv Project_3_data.csv Project_3_data.txt
[gairikchakraborty16gmail@ip-10-0-42-218 Project_data]$ ls
Project_3_data.txt
[gairikchakraborty16gmail@ip-10-0-42-218 Project_data]$ hdfs dfs -ls
Found 7 items
drwx------   - gairikchakraborty16gmail hadoop          0 2022-04-08 22:39 .Trash
drwxr-xr-x   - gairikchakraborty16gmail hadoop          0 2022-04-08 22:00 .sparkStaging
drwx------   - gairikchakraborty16gmail hadoop          0 2022-03-16 17:58 .staging
drwxr-xr-x   - gairikchakraborty16gmail hadoop          0 2022-03-15 06:07 bigdata
drwxr-xr-x   - gairikchakraborty16gmail hadoop          0 2022-04-08 22:39 project
-rw-r--r--   3 gairikchakraborty16gmail hadoop        120 2021-07-11 14:51 spark_dir
drwxr-xr-x   - gairikchakraborty16gmail hadoop          0 2021-07-11 14:44 sqoop_dir
[gairikchakraborty16gmail@ip-10-0-42-218 Project_data]$ hdfs dfs -ls project
[gairikchakraborty16gmail@ip-10-0-42-218 Project_data]$ hdfs dfs -put Project_3_data.txt project
[gairikchakraborty16gmail@ip-10-0-42-218 Project_data]$ hdfs dfs -ls project
Found 1 items
-rw-r--r--   3 gairikchakraborty16gmail hadoop    5650234 2022-04-08 22:45 project/Project_3_data.txt
[gairikchakraborty16gmail@ip-10-0-42-218 Project_data]$
```

```
scala> var temprdd = sc.textFile("project/Project_3_data.txt")
temprdd: org.apache.spark.rdd.RDD[String] = project/Project_3_data.txt MapPartitionsRDD[384] at textFile at <console>:24

scala> temprdd.map(input=>input.replace("""", "")).coalesce(1).saveAsTextFile("project/cleanData/"
     | )

scala>
```

```
[gairikchakraborty16gmail@ip-10-0-42-218 Project_data]$ hdfs dfs -ls project/cleanData
Found 2 items
-rw-r--r--   3 gairikchakraborty16gmail hadoop          0 2022-04-08 22:47 project/cleanData/_SUCCESS
-rw-r--r--   3 gairikchakraborty16gmail hadoop    3706094 2022-04-08 22:47 project/cleanData/part-00000
[gairikchakraborty16gmail@ip-10-0-42-218 Project_data]$
```

```
ip-10-0-42-218 login: gairikchakraborty16gmail
Password:
Last login: Fri Apr  8 15:03:11 on pts/659
[gairikchakraborty16gmail@ip-10-0-42-218 ~]$ hdfs dfs -ls
Found 6 items
drwx------   - gairikchakraborty16gmail hadoop          0 2022-03-16 08:00 .Trash
drwxr-xr-x   - gairikchakraborty16gmail hadoop          0 2022-04-08 09:00 .sparkStaging
drwx------   - gairikchakraborty16gmail hadoop          0 2022-03-16 17:58 .staging
drwxr-xr-x   - gairikchakraborty16gmail hadoop          0 2022-03-15 06:07 bigdata
-rw-r--r--   3 gairikchakraborty16gmail hadoop        120 2021-07-11 14:51 spark_dir
drwxr-xr-x   - gairikchakraborty16gmail hadoop          0 2021-07-11 14:44 sqoop_dir
[gairikchakraborty16gmail@ip-10-0-42-218 ~]$ hdfs dfs -mkdir project
[gairikchakraborty16gmail@ip-10-0-42-218 ~]$ hdfs dfs -ls
Found 7 items
drwx------   - gairikchakraborty16gmail hadoop          0 2022-03-16 08:00 .Trash
drwxr-xr-x   - gairikchakraborty16gmail hadoop          0 2022-04-08 09:00 .sparkStaging
drwx------   - gairikchakraborty16gmail hadoop          0 2022-03-16 17:58 .staging
drwxr-xr-x   - gairikchakraborty16gmail hadoop          0 2022-03-15 06:07 bigdata
drwxr-xr-x   - gairikchakraborty16gmail hadoop          0 2022-04-08 15:12 project
-rw-r--r--   3 gairikchakraborty16gmail hadoop        120 2021-07-11 14:51 spark_dir
drwxr-xr-x   - gairikchakraborty16gmail hadoop          0 2021-07-11 14:44 sqoop_dir
[gairikchakraborty16gmail@ip-10-0-42-218 ~]$ hdfs dfs -ls project
[gairikchakraborty16gmail@ip-10-0-42-218 ~]$ hdfs dfs -put Project_data/project_3_data.txt project
[gairikchakraborty16gmail@ip-10-0-42-218 ~]$ hdfs dfs -ls project
Found 1 items
-rw-r--r--   3 gairikchakraborty16gmail hadoop    3751306 2022-04-08 15:15 project/project_3_data.txt
[gairikchakraborty16gmail@ip-10-0-42-218 ~]$ head project/project_3_data.txt
head: cannot open ‘project/project_3_data.txt’ for reading: No such file or directory
[gairikchakraborty16gmail@ip-10-0-42-218 ~]$ cat project/project_3_data.txt
cat: project/project_3_data.txt: No such file or directory
[gairikchakraborty16gmail@ip-10-0-42-218 ~]$ hdfs dfs -ls project
Found 1 items
-rw-r--r--   3 gairikchakraborty16gmail hadoop    3751306 2022-04-08 15:15 project/project_3_data.txt
[gairikchakraborty16gmail@ip-10-0-42-218 ~]$ hdfs dfs -cat project/project_3_data.txt
age,job,marital,education,default,balance,housing,loan,contact,day,month,duration,campaign,pdays,previous,poutcome,y
58,management,married,tertiary,no,2143,yes,no,unknown,5,may,261,1,-1,0,unknown,no
44,technician,single,secondary,no,29,yes,no,unknown,5,may,151,1,-1,0,unknown,no
33,entrepreneur,married,secondary,no,2,yes,yes,unknown,5,may,76,1,-1,0,unknown,no
47,blue-collar,married,unknown,no,1506,yes,no,unknown,5,may,92,1,-1,0,unknown,no
33,unknown,single,unknown,no,1,no,no,unknown,5,may,198,1,-1,0,unknown,no
35,management,married,tertiary,no,231,yes,no,unknown,5,may,139,1,-1,0,unknown,no
```

Q.1. Load data and create a Spark data frame.

- Open Spark shell, create a RDD to store the dataset.
- Create a Dataframe from the RDD using **Spark Reader API**.

```
scala> var drdd = sc.textFile("project/project_3_data.txt")
drdd: org.apache.spark.rdd.RDD[String] = project/project_3_data.txt MapPartitionsRDD[1] at textFile at <console>:24

scala> drdd.collect
res0: Array[String] = Array(age,job,marital,education,default,balance,housing,loan,contact,day,month,duration,campaign,pdays,previous,poutcome,y, 58,management,married,tertiary,no,2143
,yes,no,unknown,5,may,261,1,-1,0,unknown,no, 44,technician,single,secondary,no,29,yes,no,unknown,5,may,151,1,-1,0,unknown,no, 33,entrepreneur,married,secondary,no,2,yes,yes,unknown,5,m
ay,76,1,-1,0,unknown,no, 47,blue-collar,married,unknown,no,1506,yes,no,unknown,5,may,92,1,-1,0,unknown,no, 33,unknown,single,unknown,no,1,no,no,unknown,5,may,198,1,-1,0,unknown,no, 35,
management,married,tertiary,no,231,yes,no,unknown,5,may,139,1,-1,0,unknown,no, 28,management,single,tertiary,no,447,yes,yes,unknown,5,may,217,1,-1,0,unknown,no, 42,entrepreneur,divorce
d,tertiary,yes,2,yes,no,unknown,5,may,380,1,-1,0,unknown,no, ...
scala> var df = spark.read.format("csv").option("header","true").option("delimiter",",").option("inferSchema","true").load("project/project_3_data.txt")
22/04/08 15:25:29 WARN lineage.LineageWriter: Lineage directory /var/log/spark/lineage doesn't exist or is not writable. Lineage for this application will be disabled.
df: org.apache.spark.sql.DataFrame = [age: int, job: string ... 15 more fields]

scala> df.show
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+-----+--------+--------+---+
|age|        job| marital|education|default|balance|housing|loan|contact|day|month|duration|campaign|pdays|previous|poutcome|  y|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+-----+--------+--------+---+
| 58|  management| married| tertiary|     no|   2143|    yes|  no|unknown|  5|  may|     261|       1|   -1|       0| unknown| no|
| 44|  technician|  single|secondary|     no|     29|    yes|  no|unknown|  5|  may|     151|       1|   -1|       0| unknown| no|
| 33|entrepreneur| married|secondary|     no|      2|    yes| yes|unknown|  5|  may|      76|       1|   -1|       0| unknown| no|
| 47| blue-collar| married|  unknown|     no|   1506|    yes|  no|unknown|  5|  may|      92|       1|   -1|       0| unknown| no|
| 33|     unknown|  single|  unknown|     no|      1|     no|  no|unknown|  5|  may|     198|       1|   -1|       0| unknown| no|
| 35|  management| married| tertiary|     no|    231|    yes|  no|unknown|  5|  may|     139|       1|   -1|       0| unknown| no|
| 28|  management|  single| tertiary|     no|    447|    yes| yes|unknown|  5|  may|     217|       1|   -1|       0| unknown| no|
| 42|entrepreneur|divorced| tertiary|    yes|      2|    yes|  no|unknown|  5|  may|     380|       1|   -1|       0| unknown| no|
| 58|     retired| married|  primary|     no|    121|    yes|  no|unknown|  5|  may|      50|       1|   -1|       0| unknown| no|
| 43|  technician|  single|secondary|     no|    593|    yes|  no|unknown|  5|  may|      55|       1|   -1|       0| unknown| no|
| 41|      admin.|divorced|secondary|     no|    270|    yes|  no|unknown|  5|  may|     222|       1|   -1|       0| unknown| no|
| 29|      admin.|  single|secondary|     no|    390|    yes|  no|unknown|  5|  may|     137|       1|   -1|       0| unknown| no|
| 53|  technician| married|secondary|     no|      6|    yes|  no|unknown|  5|  may|     517|       1|   -1|       0| unknown| no|
| 58|  technician| married|  unknown|     no|     71|    yes|  no|unknown|  5|  may|      71|       1|   -1|       0| unknown| no|
| 57|    services| married|secondary|     no|    162|    yes|  no|unknown|  5|  may|     174|       1|   -1|       0| unknown| no|
| 51|     retired| married|  primary|     no|    229|    yes|  no|unknown|  5|  may|     353|       1|   -1|       0| unknown| no|
| 45|      admin.|  single|  unknown|     no|     13|    yes|  no|unknown|  5|  may|      98|       1|   -1|       0| unknown| no|
| 57| blue-collar| married|  primary|     no|     52|    yes|  no|unknown|  5|  may|      38|       1|   -1|       0| unknown| no|
| 60|     retired| married|  primary|     no|     60|    yes|  no|unknown|  5|  may|     219|       1|   -1|       0| unknown| no|
| 33|    services| married|secondary|     no|      0|    yes|  no|unknown|  5|  may|      54|       1|   -1|       0| unknown| no|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+-----+--------+--------+---+
only showing top 20 rows
```

Q.2. Give marketing success rate (No. of people subscribed / total no. of entries) and marketing failure rate.

- Create a filter with **y=no** and count the same, and divide it by the total no. of entries to take out the marketing failure rate. Same should be done for taking out the marketing success rate with **y=yes.**

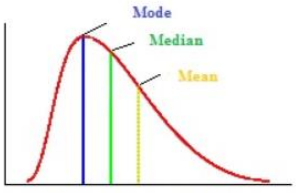
```
scala> df.show
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+-----+--------+--------+---+
|age|        job| marital|education|default|balance|housing|loan|contact|day|month|duration|campaign|pdays|previous|poutcome|  y|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+-----+--------+--------+---+
| 58|  management| married| tertiary|     no|   2143|    yes|  no|unknown|  5|  may|     261|       1|   -1|       0| unknown| no|
| 44|  technician|  single|secondary|     no|     29|    yes|  no|unknown|  5|  may|     151|       1|   -1|       0| unknown| no|
| 33|entrepreneur| married|secondary|     no|      2|    yes| yes|unknown|  5|  may|      76|       1|   -1|       0| unknown| no|
| 47| blue-collar| married|  unknown|     no|   1506|    yes|  no|unknown|  5|  may|      92|       1|   -1|       0| unknown| no|
| 33|     unknown|  single|  unknown|     no|      1|     no|  no|unknown|  5|  may|     198|       1|   -1|       0| unknown| no|
| 35|  management| married| tertiary|     no|    231|    yes|  no|unknown|  5|  may|     139|       1|   -1|       0| unknown| no|
| 28|  management|  single| tertiary|     no|    447|    yes| yes|unknown|  5|  may|     217|       1|   -1|       0| unknown| no|
| 42|entrepreneur|divorced| tertiary|    yes|      2|    yes|  no|unknown|  5|  may|     380|       1|   -1|       0| unknown| no|
| 58|     retired| married|  primary|     no|    121|    yes|  no|unknown|  5|  may|      50|       1|   -1|       0| unknown| no|
| 43|  technician|  single|secondary|     no|    593|    yes|  no|unknown|  5|  may|      55|       1|   -1|       0| unknown| no|
| 41|      admin.|divorced|secondary|     no|    270|    yes|  no|unknown|  5|  may|     222|       1|   -1|       0| unknown| no|
| 29|      admin.|  single|secondary|     no|    390|    yes|  no|unknown|  5|  may|     137|       1|   -1|       0| unknown| no|
| 53|  technician| married|secondary|     no|      6|    yes|  no|unknown|  5|  may|     517|       1|   -1|       0| unknown| no|
| 58|  technician| married|  unknown|     no|     71|    yes|  no|unknown|  5|  may|      71|       1|   -1|       0| unknown| no|
| 57|    services| married|secondary|     no|    162|    yes|  no|unknown|  5|  may|     174|       1|   -1|       0| unknown| no|
| 51|     retired| married|  primary|     no|    229|    yes|  no|unknown|  5|  may|     353|       1|   -1|       0| unknown| no|
| 45|      admin.|  single|  unknown|     no|     13|    yes|  no|unknown|  5|  may|      98|       1|   -1|       0| unknown| no|
| 57| blue-collar| married|  primary|     no|     52|    yes|  no|unknown|  5|  may|      38|       1|   -1|       0| unknown| no|
| 60|     retired| married|  primary|     no|     60|    yes|  no|unknown|  5|  may|     219|       1|   -1|       0| unknown| no|
| 33|    services| married|secondary|     no|      0|    yes|  no|unknown|  5|  may|      54|       1|   -1|       0| unknown| no|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+-----+--------+--------+---+
only showing top 20 rows


scala> df.filter("y='no').count.toFloat/df.count
<console>:1: error: unclosed string literal
df.filter("y='no').count.toFloat/df.count
          ^

scala> df.filter("y='no'").count.toFloat/df.count
res1: Float = 0.8830152

scala> df.filter("y='yes'").count.toFloat/df.count
res2: Float = 0.11698481

scala>
```

Conclusion: The marketing success rate is about **11.69%** and the failure rate is approximately **88.3%**.

Q.3. Check the quality of customers by checking average balance, median balance of customers.

- Create a new Dataframe **df_sorted** out of **df** which is sorted by the column "**balance**".
- Select the column "**balance**" and apply aggregation **mean** on this column to find out the average balance of customer's bank account.
- Select the column "**balance**" and fetch the value of balance from the middlemost row of the Dataframe **df_sorted** which is inturn the median balance of customer's bank account.

Conclusion: The quality of the customers is pretty average with average balance of **1362.27** and median balance of **448**. The stark difference between the average and median balance states that there is a wide line between the economic conditions of the bank customers (Right skewed/Positive skewness), there are more no. of customers with balance on the higher side which settles the median balance at 448, but there are less no. of customers with huge balance which rounds off the average balance at 1362.



Right-Skewed (Positive Skewness)

```
scala> var df_sorted = df.orderBy("balance")
df_sorted: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [age: int, job: string ... 15 more fields]

scala> df_sorted.show
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
|age|          job| marital|education|default|balance|housing|loan|  contact|day|month|duration|campaign|pdays|previous|poutcome|  y|
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
| 26|  blue-collar|  single|secondary|    yes|  -8019|     no| yes| cellular|  7|  jul|     299|       3|   -1|       0| unknown| no|
| 49|   management| married| tertiary|    yes|  -6847|     no| yes| cellular| 21|  jul|     206|       1|   -1|       0| unknown| no|
| 60|   management|divorced| tertiary|     no|  -4057|    yes|  no| cellular| 18|  may|     242|       6|   -1|       0| unknown| no|
| 43|   management| married| tertiary|    yes|  -3372|    yes|  no|  unknown| 29|  may|     386|       2|   -1|       0| unknown| no|
| 57|self-employed| married| tertiary|    yes|  -3313|    yes| yes|  unknown|  9|  may|     153|       1|   -1|       0| unknown| no|
| 39|self-employed| married| tertiary|     no|  -3058|    yes| yes| cellular| 17|  apr|     882|       3|   -1|       0| unknown|yes|
| 40|   technician| married| tertiary|    yes|  -2827|    yes| yes| cellular| 31|  jul|     843|       1|   -1|       0| unknown| no|
| 52|   management| married| tertiary|     no|  -2712|    yes| yes| cellular|  2|  apr|     253|       1|   -1|       0| unknown| no|
| 49|  blue-collar|  single|  primary|    yes|  -2604|    yes|  no| cellular| 18|  nov|     142|       1|   -1|       0| unknown| no|
| 51|   management|divorced| tertiary|     no|  -2282|    yes| yes| cellular| 14|  jul|     301|       6|   -1|       0| unknown| no|
| 43|     services| married|  primary|     no|  -2122|    yes| yes| cellular| 18|  nov|     141|       3|   -1|       0| unknown| no|
| 38|  blue-collar|divorced|secondary|     no|  -2093|    yes| yes|  unknown|  9|  jul|     120|       3|   -1|       0| unknown| no|
| 51| entrepreneur| married|secondary|    yes|  -2082|     no| yes| cellular| 28|  jul|     123|       6|   -1|       0| unknown| no|
| 49|   management|divorced| tertiary|     no|  -2049|    yes|  no|  unknown| 30|  may|     169|       3|   -1|       0| unknown| no|
| 35|   management|  single| tertiary|    yes|  -1980|    yes| yes| cellular| 11|  aug|     227|       1|   -1|       0| unknown| no|
| 56|   management|divorced| tertiary|    yes|  -1968|     no|  no|  unknown| 20|  jun|      60|       3|   -1|       0| unknown| no|
| 49| entrepreneur| married|secondary|     no|  -1965|     no| yes|telephone| 10|  jul|     317|       2|   -1|       0| unknown| no|
| 51|   technician| married|secondary|     no|  -1944|    yes|  no| cellular|  7|  may|     623|       1|   -1|       0| unknown|yes|
| 36|    housemaid| married| tertiary|    yes|  -1941|    yes|  no|  unknown| 16|  jun|     505|       1|   -1|       0| unknown| no|
| 37|   management| married| tertiary|     no|  -1884|    yes|  no| cellular| 21|  jul|     193|       1|   -1|       0| unknown| no|
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
only showing top 20 rows


scala> df_sorted.select("balance").agg(avg("balance")).show
+------------------+
|      avg(balance)|
+------------------+
|1362.2720576850766|
+------------------+


scala> df_sorted.select("balance").collect()(df_sorted.count.toInt/2)
res14: org.apache.spark.sql.Row = [448]
```

Q.4. Maximum, Mean, and Minimum age of average targeted customers.

- Select the column "**age**" and apply aggregation **mean, minimum** and **maximum** on this column to find out the maximum, mean, and minimum age of average targeted customers.

```
| 49|  blue-collar|  single|  primary|    yes|  -2604|    yes|  no| cellular| 18|  nov|     142|       1|   -1|       0| unknown| no|
| 51|   management|divorced| tertiary|     no|  -2282|    yes| yes| cellular| 14|  jul|     301|       6|   -1|       0| unknown| no|
| 43|     services| married|  primary|     no|  -2122|    yes| yes| cellular| 18|  nov|     141|       3|   -1|       0| unknown| no|
| 38|  blue-collar|divorced|secondary|     no|  -2093|    yes| yes|  unknown|  9|  jul|     120|       3|   -1|       0| unknown| no|
| 51| entrepreneur| married|secondary|    yes|  -2082|     no| yes| cellular| 28|  jul|     123|       6|   -1|       0| unknown| no|
| 49|   management|divorced| tertiary|     no|  -2049|    yes|  no|  unknown| 30|  may|     169|       3|   -1|       0| unknown| no|
| 35|   management|  single| tertiary|    yes|  -1980|    yes| yes| cellular| 11|  aug|     227|       1|   -1|       0| unknown| no|
| 56|   management|divorced| tertiary|    yes|  -1968|     no|  no|  unknown| 20|  jun|      60|       3|   -1|       0| unknown| no|
| 49| entrepreneur| married|secondary|     no|  -1965|     no| yes|telephone| 10|  jul|     317|       2|   -1|       0| unknown| no|
| 51|   technician| married|secondary|     no|  -1944|    yes|  no| cellular|  7|  may|     623|       1|   -1|       0| unknown|yes|
| 36|    housemaid| married| tertiary|    yes|  -1941|    yes|  no|  unknown| 16|  jun|     505|       1|   -1|       0| unknown| no|
| 37|   management| married| tertiary|     no|  -1884|    yes|  no| cellular| 21|  jul|     193|       1|   -1|       0| unknown| no|
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
only showing top 20 rows


scala> df_sorted.select("age").agg(round(avg("age"),0)).show
+------------------+
|round(avg(age), 0)|
+------------------+
|              41.0|
+------------------+


scala> df_sorted.select("age").agg(min("age")).show
+--------+
|min(age)|
+--------+
|      18|
+--------+


scala> df_sorted.select("age").agg(max("age")).show
+--------+
|max(age)|
+--------+
|      95|
+--------+


scala>
```

Conclusion: For the average targeted customers, the mean age is **41**, the maximum age is **95** and lastly, the minimum age is **18**.

Q.5. Check if age matters in marketing subscription for deposit.

- Apply **groupby** method on the Dataframe **df_sorted** w.r.t. the column "**y**" and apply aggregation **mean** on the "**age**" column to find out the average age of customers who have or haven't subscribed to the bank term deposit.

```
scala> df_sorted.show
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
|age|          job| marital|education|default|balance|housing|loan|  contact|day|month|duration|campaign|pdays|previous|poutcome|  y|
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
| 26|  blue-collar|  single|secondary|    yes|  -8019|     no| yes| cellular|  7|  jul|     299|       3|   -1|       0| unknown| no|
| 49|   management| married| tertiary|    yes|  -6847|     no| yes| cellular| 21|  jul|     206|       1|   -1|       0| unknown| no|
| 60|   management|divorced| tertiary|     no|  -4057|    yes|  no| cellular| 18|  may|     242|       6|   -1|       0| unknown| no|
| 43|   management| married| tertiary|    yes|  -3372|    yes|  no|  unknown| 29|  may|     386|       2|   -1|       0| unknown| no|
| 57|self-employed| married| tertiary|    yes|  -3313|    yes| yes|  unknown|  9|  may|     153|       1|   -1|       0| unknown| no|
| 39|self-employed| married| tertiary|     no|  -3058|    yes| yes| cellular| 17|  apr|     882|       3|   -1|       0| unknown|yes|
| 40|   technician| married| tertiary|    yes|  -2827|    yes| yes| cellular| 31|  jul|     843|       1|   -1|       0| unknown| no|
| 52|   management| married| tertiary|     no|  -2712|    yes| yes| cellular|  2|  apr|     253|       1|   -1|       0| unknown| no|
| 49|  blue-collar|  single|  primary|    yes|  -2604|    yes|  no| cellular| 18|  nov|     142|       1|   -1|       0| unknown| no|
| 51|   management|divorced| tertiary|     no|  -2282|    yes| yes| cellular| 14|  jul|     301|       6|   -1|       0| unknown| no|
| 43|     services| married|  primary|     no|  -2122|    yes| yes| cellular| 18|  nov|     141|       3|   -1|       0| unknown| no|
| 38|  blue-collar|divorced|secondary|     no|  -2093|    yes| yes|  unknown|  9|  jul|     120|       3|   -1|       0| unknown| no|
| 51| entrepreneur| married|secondary|    yes|  -2082|     no| yes| cellular| 28|  jul|     123|       6|   -1|       0| unknown| no|
| 49|   management|divorced| tertiary|     no|  -2049|    yes|  no|  unknown| 30|  may|     169|       3|   -1|       0| unknown| no|
| 35|   management|  single| tertiary|    yes|  -1980|    yes| yes| cellular| 11|  aug|     227|       1|   -1|       0| unknown| no|
| 56|   management|divorced| tertiary|    yes|  -1968|     no|  no|  unknown| 20|  jun|      60|       3|   -1|       0| unknown| no|
| 49| entrepreneur| married|secondary|     no|  -1965|     no| yes|telephone| 10|  jul|     317|       2|   -1|       0| unknown| no|
| 51|   technician| married|secondary|     no|  -1944|    yes|  no| cellular|  7|  may|     623|       1|   -1|       0| unknown|yes|
| 36|    housemaid| married| tertiary|    yes|  -1941|    yes|  no|  unknown| 16|  jun|     505|       1|   -1|       0| unknown| no|
| 37|   management| married| tertiary|     no|  -1884|    yes|  no| cellular| 21|  jul|     193|       1|   -1|       0| unknown| no|
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
only showing top 20 rows


scala> df_sorted.groupBy("y").agg(round(avg("age"),0)as("Average age")).show
+---+-----------+
|  y|Average age|
+---+-----------+
| no|       41.0|
|yes|       42.0|
+---+-----------+


scala>
```

Conclusion: The average age of customers who have or haven't subscribed to the bank term deposit are **42** and **41** respectively. The values are pretty much same with a mere difference of 1 which confirms that the subscribers of this campaign belong to different age groups and there is no direct relation between age of the customers and whether they have subscribed or not.

Q.6. Check if marital status mattered for subscription to deposit.

- Apply **groupby** method on the Dataframe **df_sorted** w.r.t the columns "**y**" and "**marital**" and apply aggregation **count** the total number of entries to estimate the relationship between the marital status of the customers and whether they have subscribed for the bank term deposit or not.

Conclusion: The estimate clearly shows that half of the customers who haven't subscribed for the scheme are married, followed by the one-fifth of customers who are single and didn't subscribe to the scheme. Approximately, one-tenth of the customers are those who got divorced and didn't subscribe. The customers who subscribed to the scheme are very minimal and the number is equally distributed among different marital status, which is about one-eighteenth of the total customers each.

```
scala> df_sorted.show
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
|age|          job| marital|education|default|balance|housing|loan|  contact|day|month|duration|campaign|pdays|previous|poutcome|  y|
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
| 26|  blue-collar|  single|secondary|    yes|  -8019|     no| yes| cellular|  7|  jul|     299|       3|   -1|       0| unknown| no|
| 49|   management| married| tertiary|    yes|  -6847|     no| yes| cellular| 21|  jul|     206|       1|   -1|       0| unknown| no|
| 60|   management|divorced| tertiary|     no|  -4057|    yes|  no| cellular| 18|  may|     242|       6|   -1|       0| unknown| no|
| 43|   management| married| tertiary|    yes|  -3372|    yes|  no|  unknown| 29|  may|     386|       2|   -1|       0| unknown| no|
| 57|self-employed| married| tertiary|    yes|  -3313|    yes| yes|  unknown|  9|  may|     153|       1|   -1|       0| unknown| no|
| 39|self-employed| married| tertiary|     no|  -3058|    yes| yes| cellular| 17|  apr|     882|       3|   -1|       0| unknown|yes|
| 40|   technician| married| tertiary|    yes|  -2827|    yes| yes| cellular| 31|  jul|     843|       1|   -1|       0| unknown| no|
| 52|   management| married| tertiary|     no|  -2712|    yes| yes| cellular|  2|  apr|     253|       1|   -1|       0| unknown| no|
| 49|  blue-collar|  single|  primary|    yes|  -2604|    yes|  no| cellular| 18|  nov|     142|       1|   -1|       0| unknown| no|
| 51|   management|divorced| tertiary|     no|  -2282|    yes| yes| cellular| 14|  jul|     301|       6|   -1|       0| unknown| no|
| 43|     services| married|  primary|     no|  -2122|    yes| yes| cellular| 18|  nov|     141|       3|   -1|       0| unknown| no|
| 38|  blue-collar|divorced|secondary|     no|  -2093|    yes| yes|  unknown|  9|  jul|     120|       3|   -1|       0| unknown| no|
| 51| entrepreneur| married|secondary|    yes|  -2082|     no| yes| cellular| 28|  jul|     123|       6|   -1|       0| unknown| no|
| 49|   management|divorced| tertiary|     no|  -2049|    yes|  no|  unknown| 30|  may|     169|       3|   -1|       0| unknown| no|
| 35|   management|  single| tertiary|    yes|  -1980|    yes| yes| cellular| 11|  aug|     227|       1|   -1|       0| unknown| no|
| 56|   management|divorced| tertiary|    yes|  -1968|     no|  no|  unknown| 20|  jun|      60|       3|   -1|       0| unknown| no|
| 49| entrepreneur| married|secondary|     no|  -1965|     no| yes|telephone| 10|  jul|     317|       2|   -1|       0| unknown| no|
| 51|   technician| married|secondary|     no|  -1944|    yes|  no| cellular|  7|  may|     623|       1|   -1|       0| unknown|yes|
| 36|    housemaid| married| tertiary|    yes|  -1941|    yes|  no|  unknown| 16|  jun|     505|       1|   -1|       0| unknown| no|
| 37|   management| married| tertiary|     no|  -1884|    yes|  no| cellular| 21|  jul|     193|       1|   -1|       0| unknown| no|
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
only showing top 20 rows


scala> df_sorted.groupBy("y","marital").agg(count("*")as("No. of customers")).show
+---+--------+---------------+
|  y| marital|No. of customers|
+---+--------+---------------+
| no| married|          24459|
| no|  single|          10878|
|yes|  single|           1912|
|yes| married|           2755|
| no|divorced|           4585|
|yes|divorced|            622|
+---+--------+---------------+
```

Q.7. Check if age and marital status together mattered for subscription to deposit scheme.

- Apply **groupby** method on the Dataframe **df_sorted** w.r.t the columns "**y**" and "**marital**" and apply aggregation **mean** on "**age**" column and **count** the total number of entries to estimate the relationship between the marital status of the customers along with their age and whether they have subscribed for the bank term deposit or not.

```
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
|age|          job| marital|education|default|balance|housing|loan|  contact|day|month|duration|campaign|pdays|previous|poutcome|  y|
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
| 26|  blue-collar|  single|secondary|    yes|  -8019|     no| yes| cellular|  7|  jul|     299|       3|   -1|       0| unknown| no|
| 49|   management| married| tertiary|    yes|  -6847|     no| yes| cellular| 21|  jul|     206|       1|   -1|       0| unknown| no|
| 60|   management|divorced| tertiary|     no|  -4057|    yes|  no| cellular| 18|  may|     242|       6|   -1|       0| unknown| no|
| 43|   management| married| tertiary|    yes|  -3372|    yes|  no|  unknown| 29|  may|     386|       2|   -1|       0| unknown| no|
| 57|self-employed| married| tertiary|    yes|  -3313|    yes| yes|  unknown|  9|  may|     153|       1|   -1|       0| unknown| no|
| 39|self-employed| married| tertiary|     no|  -3058|    yes| yes| cellular| 17|  apr|     882|       3|   -1|       0| unknown|yes|
| 40|   technician| married| tertiary|    yes|  -2827|    yes| yes| cellular| 31|  jul|     843|       1|   -1|       0| unknown| no|
| 52|   management| married| tertiary|     no|  -2712|    yes| yes| cellular|  2|  apr|     253|       1|   -1|       0| unknown| no|
| 49|  blue-collar|  single|  primary|    yes|  -2604|    yes|  no| cellular| 18|  nov|     142|       1|   -1|       0| unknown| no|
| 51|   management|divorced| tertiary|     no|  -2282|    yes| yes| cellular| 14|  jul|     301|       6|   -1|       0| unknown| no|
| 43|     services| married|  primary|     no|  -2122|    yes| yes| cellular| 18|  nov|     141|       3|   -1|       0| unknown| no|
| 38|  blue-collar|divorced|secondary|     no|  -2093|    yes| yes|  unknown|  9|  jul|     120|       3|   -1|       0| unknown| no|
| 51| entrepreneur| married|secondary|    yes|  -2082|     no| yes| cellular| 28|  jul|     123|       6|   -1|       0| unknown| no|
| 49|   management|divorced| tertiary|     no|  -2049|    yes|  no|  unknown| 30|  may|     169|       3|   -1|       0| unknown| no|
| 35|   management|  single| tertiary|    yes|  -1980|    yes| yes| cellular| 11|  aug|     227|       1|   -1|       0| unknown| no|
| 56|   management|divorced| tertiary|    yes|  -1968|     no|  no|  unknown| 20|  jun|      60|       3|   -1|       0| unknown| no|
| 49| entrepreneur| married|secondary|     no|  -1965|     no| yes|telephone| 10|  jul|     317|       2|   -1|       0| unknown| no|
| 51|   technician| married|secondary|     no|  -1944|    yes|  no| cellular|  7|  may|     623|       1|   -1|       0| unknown|yes|
| 36|    housemaid| married| tertiary|    yes|  -1941|    yes|  no|  unknown| 16|  jun|     505|       1|   -1|       0| unknown| no|
| 37|   management| married| tertiary|     no|  -1884|    yes|  no| cellular| 21|  jul|     193|       1|   -1|       0| unknown| no|
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+
only showing top 20 rows


scala> df_sorted.groupBy("y","marital").agg(round(avg("age"),0)as("Average Age"),count("*")as("No. of customers")).show
+---+--------+-----------+---------------+
|  y| marital|Average Age|No. of customers|
+---+--------+-----------+---------------+
| no| married|       43.0|          24459|
| no|  single|       34.0|          10878|
|yes|  single|       32.0|           1912|
|yes| married|       47.0|           2755|
| no|divorced|       45.0|           4585|
|yes|divorced|       49.0|            622|
+---+--------+-----------+---------------+


scala>
```

Conclusion: The estimate clearly shows that half of the customers who haven't subscribed for the scheme are married with an average age of **43**, followed by the one-fifth of customers who are single and didn't subscribe to the scheme and their average age is **34**. Approximately, one-tenth of the customers are those who got divorced and didn't subscribe have their average age to be **45**. The customers who subscribed to the scheme are very minimal and the number is equally distributed among different marital status, which is about one-eighteenth of the total customers each and have a varied average age which lies in the range of 32-49.

Q.8. Do feature engineering for column—age and find right age effect on campaign.

- Create a new Dataframe **df_new** out of **df_sorted** which introduced a new column "**Age group**" based on the age of the customer. If the customer's age is less than **30**, it is classified as "**young**", if their age is more than **30** and less than **55**, it is classified as "**middle**", else the rests are classified under "**old**".
- Apply **groupby** method on the Dataframe **df_new** w.r.t the columns "**y**" and "**Age group**" and apply aggregation **count** the total number of entries as "**No. of customers**" to estimate the relationship between the age group of the customers and whether they have subscribed for the bank term deposit or not.

```
scala> var df_new = df_sorted.withColumn("Age group",expr("case when age<30 then 'young' when age<55 then 'middle' else 'old' end"))
df_new: org.apache.spark.sql.DataFrame = [age: int, job: string ... 16 more fields]

scala> df_new.show
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+---------+
|age|          job| marital|education|default|balance|housing|loan|  contact|day|month|duration|campaign|pdays|previous|poutcome|  y|Age group|
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+---------+
| 26|  blue-collar|  single|secondary|    yes|  -8019|     no| yes| cellular|  7|  jul|     299|       3|   -1|       0| unknown| no|    young|
| 49|   management| married| tertiary|    yes|  -6847|     no| yes| cellular| 21|  jul|     206|       1|   -1|       0| unknown| no|   middle|
| 60|   management|divorced| tertiary|     no|  -4057|    yes|  no| cellular| 18|  may|     242|       6|   -1|       0| unknown| no|      old|
| 43|   management| married| tertiary|    yes|  -3372|    yes|  no|  unknown| 29|  may|     386|       2|   -1|       0| unknown| no|   middle|
| 57|self-employed| married| tertiary|    yes|  -3313|    yes| yes|  unknown|  9|  may|     153|       1|   -1|       0| unknown| no|      old|
| 39|self-employed| married| tertiary|     no|  -3058|    yes| yes| cellular| 17|  apr|     882|       3|   -1|       0| unknown|yes|   middle|
| 40|   technician| married| tertiary|    yes|  -2827|    yes| yes| cellular| 31|  jul|     843|       1|   -1|       0| unknown| no|   middle|
| 52|   management| married| tertiary|     no|  -2712|    yes| yes| cellular|  2|  apr|     253|       1|   -1|       0| unknown| no|   middle|
| 49|  blue-collar|  single|  primary|    yes|  -2604|    yes|  no| cellular| 18|  nov|     142|       1|   -1|       0| unknown| no|   middle|
| 51|   management|divorced| tertiary|     no|  -2282|    yes| yes| cellular| 14|  jul|     301|       6|   -1|       0| unknown| no|   middle|
| 43|     services| married|  primary|     no|  -2122|    yes| yes| cellular| 18|  nov|     141|       3|   -1|       0| unknown| no|   middle|
| 38|  blue-collar|divorced|secondary|     no|  -2093|    yes| yes|  unknown|  9|  jul|     120|       3|   -1|       0| unknown| no|   middle|
| 51| entrepreneur| married|secondary|    yes|  -2082|     no| yes| cellular| 28|  jul|     123|       6|   -1|       0| unknown| no|   middle|
| 49|   management|divorced| tertiary|     no|  -2049|    yes|  no|  unknown| 30|  may|     169|       3|   -1|       0| unknown| no|   middle|
| 35|   management|  single| tertiary|    yes|  -1980|    yes| yes| cellular| 11|  aug|     227|       1|   -1|       0| unknown| no|   middle|
| 56|   management|divorced| tertiary|    yes|  -1968|     no|  no|  unknown| 20|  jun|      60|       3|   -1|       0| unknown| no|      old|
| 49| entrepreneur| married|secondary|     no|  -1965|     no| yes|telephone| 10|  jul|     317|       2|   -1|       0| unknown| no|   middle|
| 51|   technician| married|secondary|     no|  -1944|    yes|  no| cellular|  7|  may|     623|       1|   -1|       0| unknown|yes|   middle|
| 36|    housemaid| married| tertiary|    yes|  -1941|    yes|  no|  unknown| 16|  jun|     505|       1|   -1|       0| unknown| no|   middle|
| 37|   management| married| tertiary|     no|  -1884|    yes|  no| cellular| 21|  jul|     193|       1|   -1|       0| unknown| no|   middle|
+---+-------------+--------+---------+-------+-------+-------+----+---------+---+-----+--------+--------+-----+--------+--------+---+---------+
only showing top 20 rows


scala> df_new.groupBy("Age group","y").agg(count("*")as("No. of customers")).show
+---------+---+---------------+
|Age group|  y|No. of customers|
+---------+---+---------------+
|   middle| no|          30853|
|   middle|yes|           3379|
|    young| no|           4345|
|      old|yes|            982|
|      old| no|           4724|
|    young|yes|            928|
```

Conclusion: The numbers clearly show that the conversion of a middle-aged customer is minimal – as only **3379** turned in and **30853** middle-aged customers opt out, thus they should be handled with extra care. For the old-aged customers, conversion is pretty much decent as **982** turned in and **4724** old-aged customers opt out. Also, for the young customers, conversion is pretty much decent as **928** turned in and **4345** young customers opt out.