Project 3 Market Analysis in Banking Domain

DESCRIPTION

Background and Objective:

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)

Dataset Description

The data fields are as follows:

1.	age	numeric
2.	job	type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
3.	marital	marital status (categorical: 'divorced', 'married', 'single', 'unknown'; note: 'divorced' means divorced or widowed)
4.	education	(categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
5.	default	has credit in default? (categorical: 'no', 'yes', 'unknown')
6.	housing:	has housing loan? (categorical: 'no', 'yes', 'unknown')
7.	loan	has a personal loan? (categorical: 'no', 'yes', 'unknown')

related to the last contact of the current campaign:

8.	contact	contact communication type (categorical: 'cellular', 'telephone')

9.	month	Month of last contact (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10.	day_of_week	last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
11.	duration	last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (example, if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call "y" is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

# other attributes:

12.	campaign	number of times a customer was contacted during the campaign (numeric, includes last contact)
13.	pdays:	number of days passed after the customer was last contacted from a previous campaign (numeric; 999 means customer was not previously contacted)
14.	previous	number of times the customer was contacted prior to (or before) this campaign (numeric)
15.	poutcome	outcome of the previous marketing campaign (categorical: 'failure', 'nonexistent', 'success')

#Output variable (desired target):

16	y	Has the customer subscribed a term deposit? (binary: 'yes', 'no')

Analysis tasks to be done-:

The data size is huge and the marketing team has asked you to perform the below analysis-

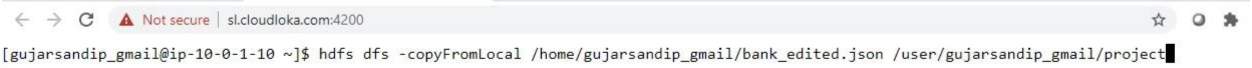
1. Load data and create a Spark data frame
2. Give marketing success rate (No. of people subscribed / total no. of entries)

- Give marketing failure rate

3. Give the maximum, mean, and minimum age of the average targeted customer
4. Check the quality of customers by checking average balance, median balance of customers
5. Check if age matters in marketing subscription for deposit
6. Check if marital status mattered for a subscription to deposit
7. Check if age and marital status together mattered for a subscription to deposit scheme
8. Do feature engineering for the bank and find the right age effect on the campaign.

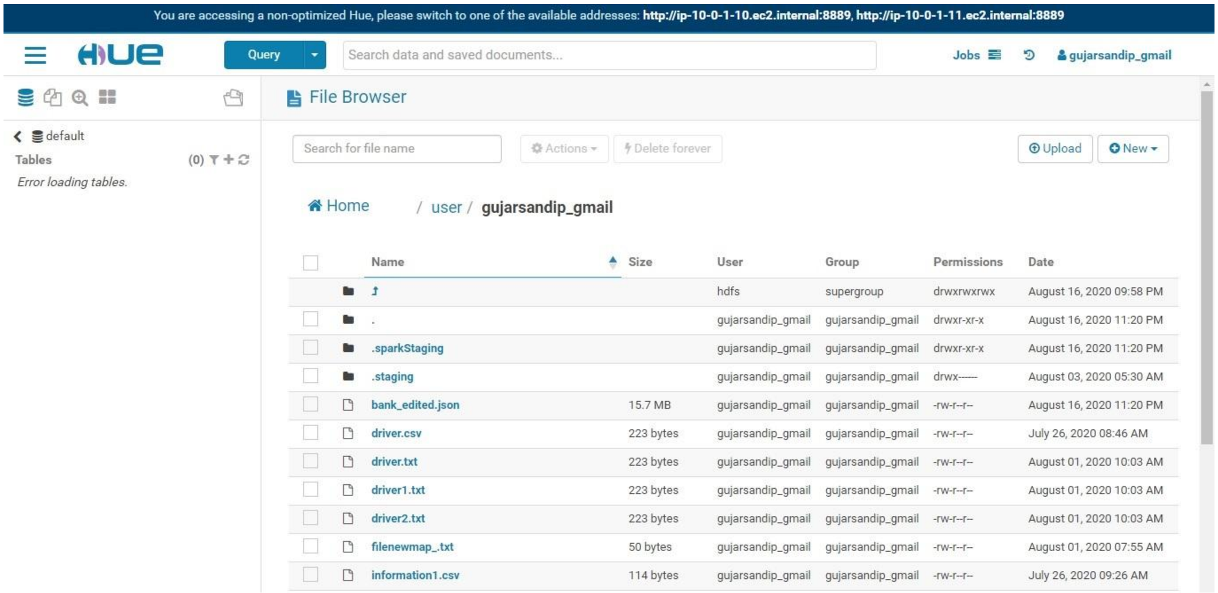
Project 3 - steps

1. Uploaded the <*bank_edited.json*> file through FTP and the copied it to HDFS with this command



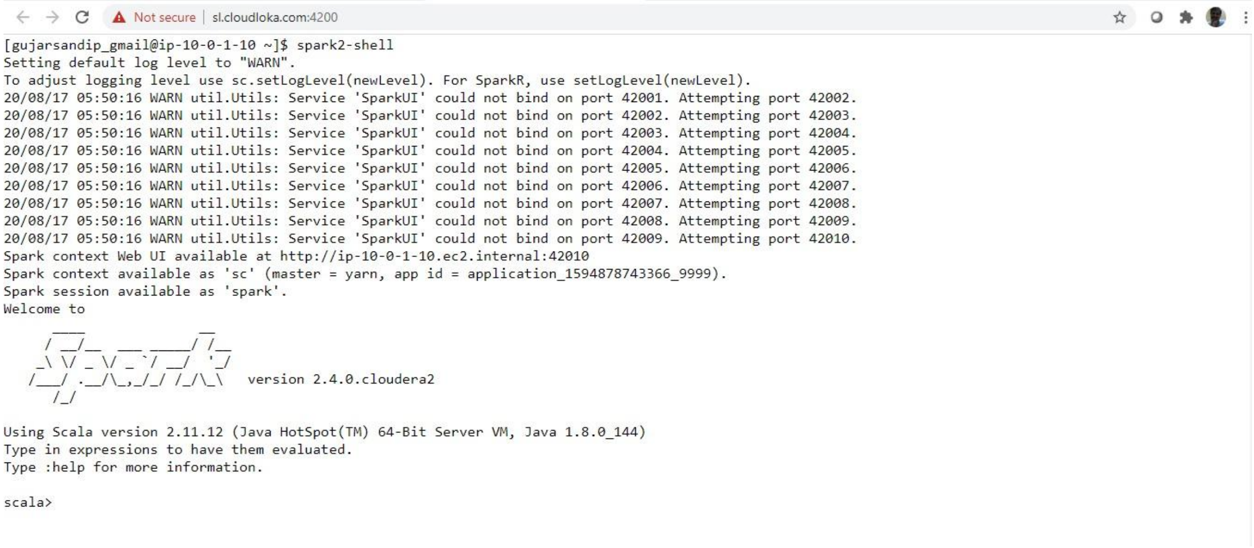
```
[gujarsandip_gmail@ip-10-0-1-10 ~]$ hdfs dfs -copyFromLocal /home/gujarsandip_gmail/bank_edited.json /user/gujarsandip_gmail/project
```

2. After this checked in Hue if the file *bank_edited.json*> is on HDFS (it is there)


You are accessing a non-optimized Hue, please switch to one of the available addresses: http://ip-10-0-1-10.ec2.internal:8889, http://ip-10-0-1-11.ec2.internal:8889
HUE
Query
Search data and saved documents...
Jobs
gujarsandip_gmail
default
Tables (0)
Error loading tables.
File Browser
Search for file name
Actions
Delete forever
Upload
New
Home / user / gujarsandip_gmail

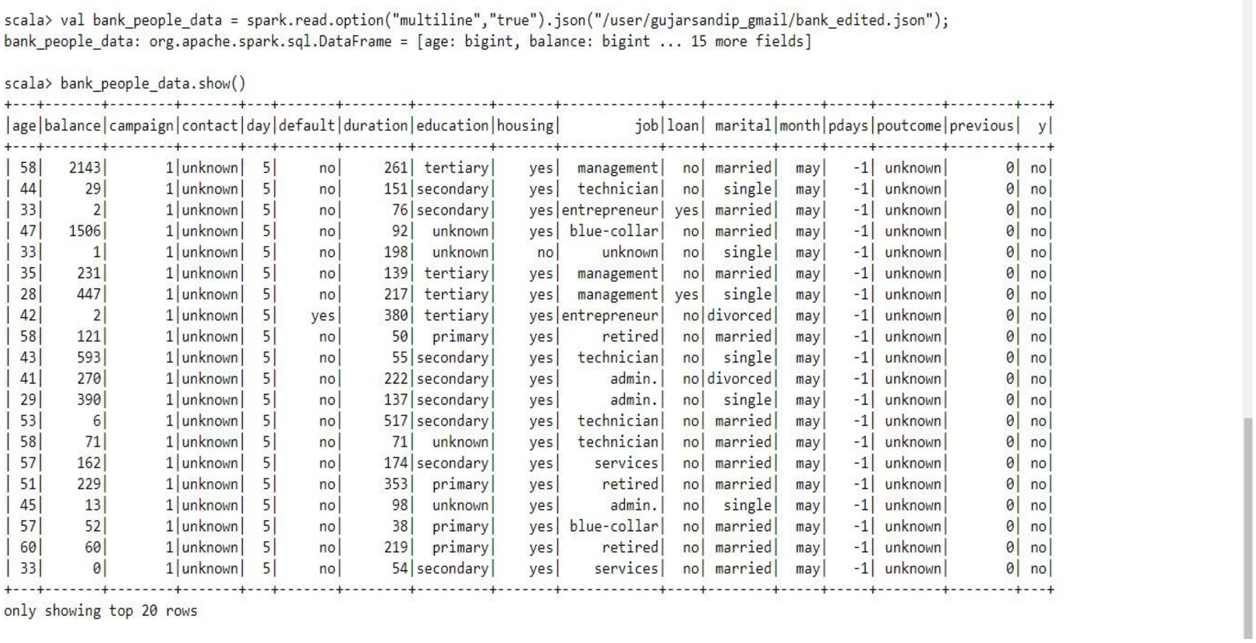
Name	Size	User	Group	Permissions	Date
↥		hdfs	supergroup	drwxrwxrwx	August 16, 2020 09:58 PM
.		gujarsandip_gmail	gujarsandip_gmail	drwxr-xr-x	August 16, 2020 11:20 PM
.sparkStaging		gujarsandip_gmail	gujarsandip_gmail	drwxr-xr-x	August 16, 2020 11:20 PM
.staging		gujarsandip_gmail	gujarsandip_gmail	drwx------	August 03, 2020 05:30 AM
bank_edited.json	15.7 MB	gujarsandip_gmail	gujarsandip_gmail	-rw-r--r--	August 16, 2020 11:20 PM
driver.csv	223 bytes	gujarsandip_gmail	gujarsandip_gmail	-rw-r--r--	July 26, 2020 08:46 AM
driver.txt	223 bytes	gujarsandip_gmail	gujarsandip_gmail	-rw-r--r--	August 01, 2020 10:03 AM
driver1.txt	223 bytes	gujarsandip_gmail	gujarsandip_gmail	-rw-r--r--	August 01, 2020 10:03 AM
driver2.txt	223 bytes	gujarsandip_gmail	gujarsandip_gmail	-rw-r--r--	August 01, 2020 10:03 AM
filenewmap_.txt	50 bytes	gujarsandip_gmail	gujarsandip_gmail	-rw-r--r--	August 01, 2020 07:55 AM
information1.csv	114 bytes	gujarsandip_gmail	gujarsandip_gmail	-rw-r--r--	July 26, 2020 09:26 AM

3. Now start Spark2 on the web console by entering spark2-shell



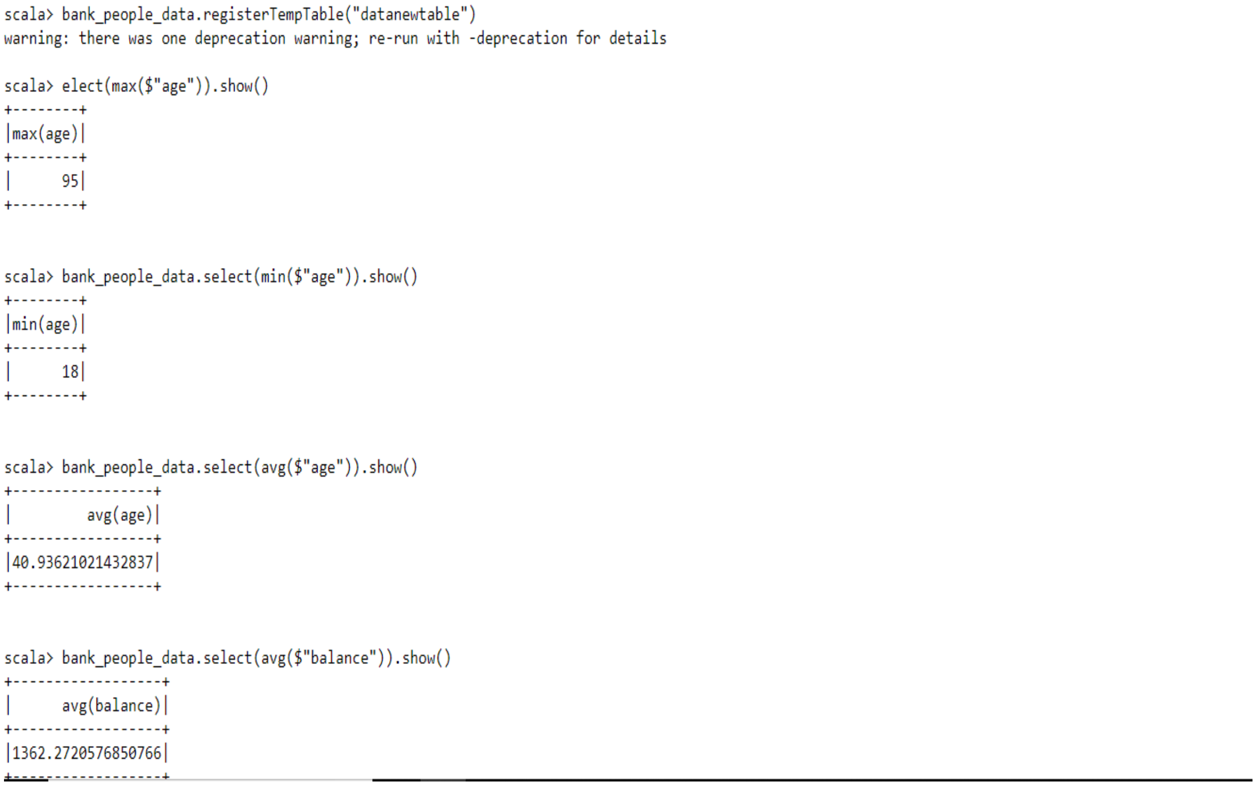
```
[gujarsandip_gmail@ip-10-0-1-10 ~]$ spark2-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
20/08/17 05:50:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42001. Attempting port 42002.
20/08/17 05:50:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42002. Attempting port 42003.
20/08/17 05:50:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42003. Attempting port 42004.
20/08/17 05:50:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42004. Attempting port 42005.
20/08/17 05:50:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42005. Attempting port 42006.
20/08/17 05:50:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42006. Attempting port 42007.
20/08/17 05:50:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42007. Attempting port 42008.
20/08/17 05:50:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42008. Attempting port 42009.
20/08/17 05:50:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42009. Attempting port 42010.
Spark context Web UI available at http://ip-10-0-1-10.ec2.internal:42010
Spark context available as 'sc' (master = yarn, app id = application_1594878743366_9999).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.0.cloudera2
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_144)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

4. The command and its output



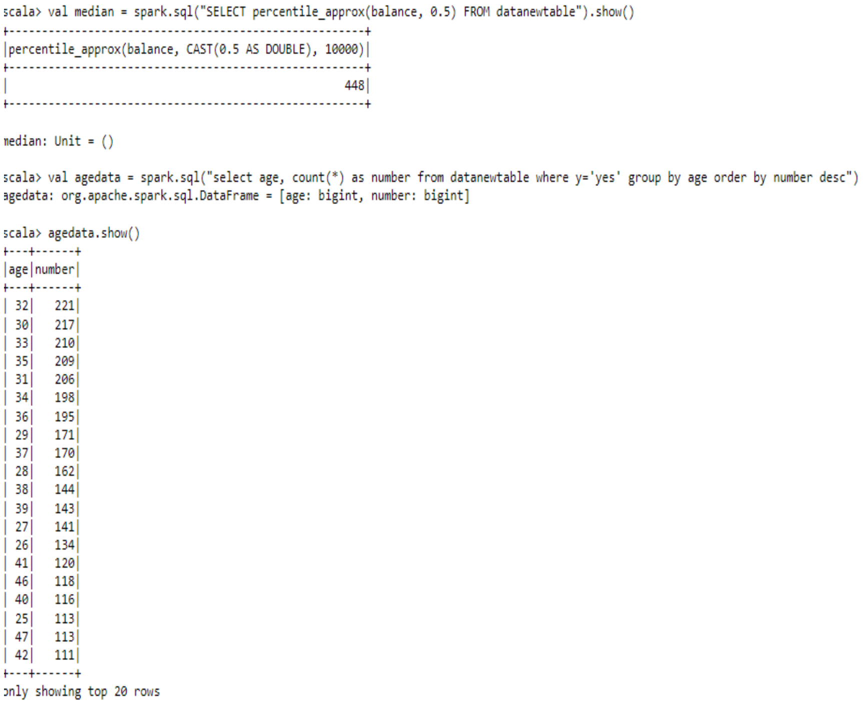
```
scala> val bank_people_data = spark.read.option("multiline","true").json("/user/gujarsandip_gmail/bank_edited.json");
bank_people_data: org.apache.spark.sql.DataFrame = [age: bigint, balance: bigint ... 15 more fields]

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

5. The command and its output

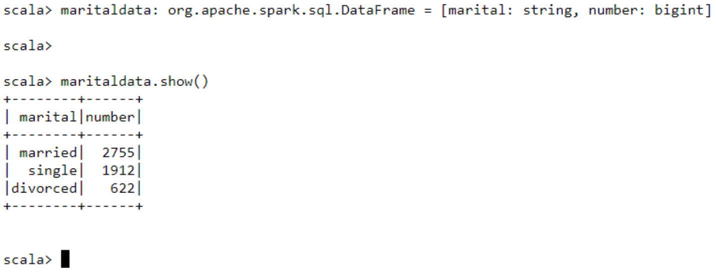


```
scala> bank_people_data.registerTempTable("datanewtable")
warning: there was one deprecation warning; re-run with -deprecation for details

scala> elect(max($"age")).show()
+--------+
|max(age)|
+--------+
|      95|
+--------+


scala> bank_people_data.select(min($"age")).show()
+--------+
|min(age)|
+--------+
|      18|
+--------+


scala> bank_people_data.select(avg($"age")).show()
+-----------------+
|         avg(age)|
+-----------------+
|40.93621021432837|
+-----------------+


scala> bank_people_data.select(avg($"balance")).show()
+------------------+
|      avg(balance)|
+------------------+
|1362.2720576850766|
+------------------+
```

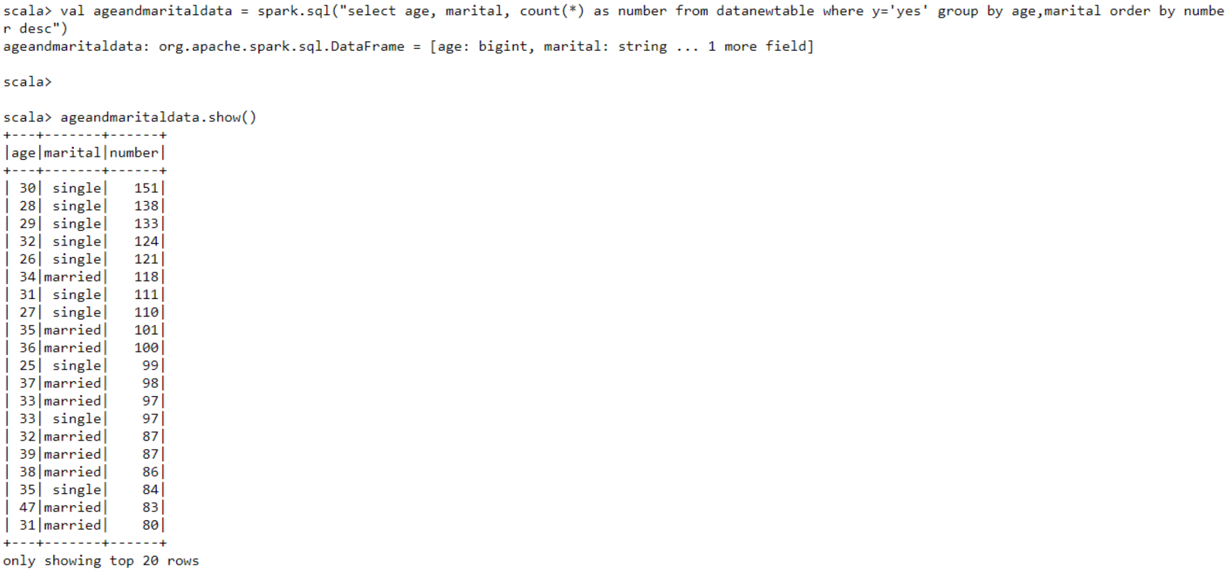
6. The command and its output



```
scala> val median = spark.sql("SELECT percentile_approx(balance, 0.5) FROM datanewtable").show()
+----------------------------------------------------+
|percentile_approx(balance, CAST(0.5 AS DOUBLE), 10000)|
+----------------------------------------------------+
|                                                 448|
+----------------------------------------------------+

median: Unit = ()

scala> val agedata = spark.sql("select age, count(*) as number from datanewtable where y='yes' group by age order by number desc")
agedata: org.apache.spark.sql.DataFrame = [age: bigint, number: bigint]

scala> agedata.show()
+---+------+
|age|number|
+---+------+
| 32|   221|
| 30|   217|
| 33|   210|
| 35|   209|
| 31|   206|
| 34|   198|
| 36|   195|
| 29|   171|
| 37|   170|
| 28|   162|
| 38|   144|
| 39|   143|
| 27|   141|
| 26|   134|
| 41|   120|
| 46|   118|
| 40|   116|
| 25|   113|
| 47|   113|
| 42|   111|
+---+------+
only showing top 20 rows
```

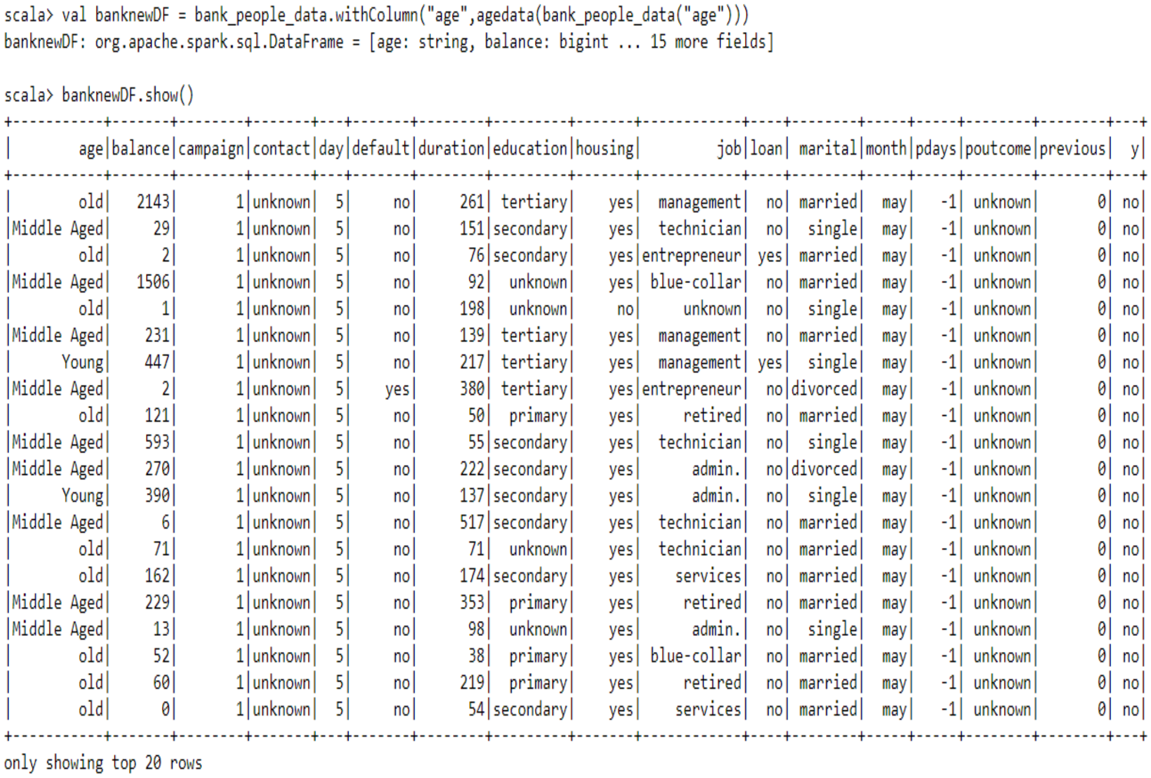


```
scala> maritaldata: org.apache.spark.sql.DataFrame = [marital: string, number: bigint]

scala>

scala> maritaldata.show()
+--------+------+
| marital|number|
+--------+------+
| married|  2755|
|  single|  1912|
|divorced|   622|
+--------+------+


scala> 
```



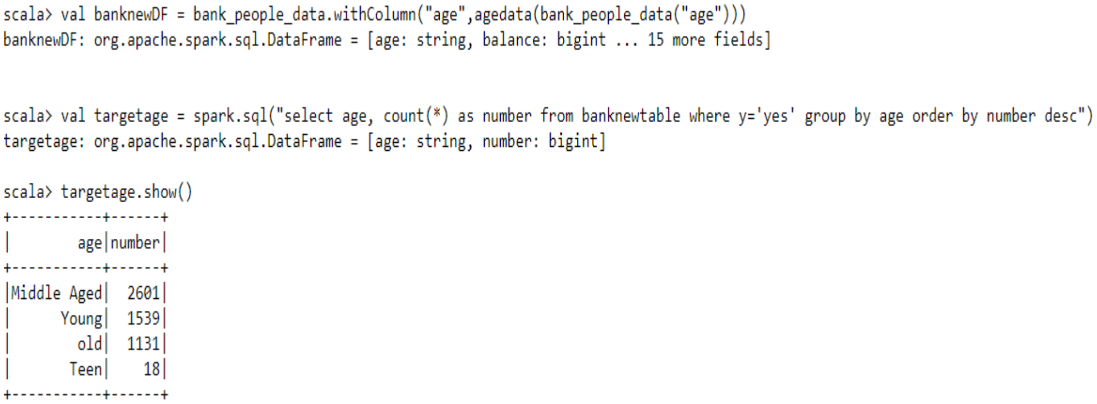
```
scala> val ageandmaritaldata = spark.sql("select age, marital, count(*) as number from datanewtable where y='yes' group by age,marital order by numbe
r desc")
ageandmaritaldata: org.apache.spark.sql.DataFrame = [age: bigint, marital: string ... 1 more field]

scala>

scala> ageandmaritaldata.show()
+---+-------+------+
|age|marital|number|
+---+-------+------+
| 30| single|   151|
| 28| single|   138|
| 29| single|   133|
| 32| single|   124|
| 26| single|   121|
| 34|married|   118|
| 31| single|   111|
| 27| single|   110|
| 35|married|   101|
| 36|married|   100|
| 25| single|    99|
| 37|married|    98|
| 33|married|    97|
| 33| single|    97|
| 32|married|    87|
| 39|married|    87|
| 38|married|    86|
| 35| single|    84|
| 47|married|    83|
| 31|married|    80|
+---+-------+------+
only showing top 20 rows
```

7. The command and its output



```
scala> val banknewDF = bank_people_data.withColumn("age",agedata(bank_people_data("age")))
banknewDF: org.apache.spark.sql.DataFrame = [age: string, balance: bigint ... 15 more fields]

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

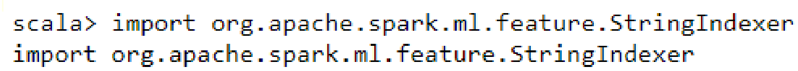
8. The command and its output



```
scala> val banknewDF = bank_people_data.withColumn("age",agedata(bank_people_data("age")))
banknewDF: org.apache.spark.sql.DataFrame = [age: string, balance: bigint ... 15 more fields]


scala> val targetage = spark.sql("select age, count(*) as number from banknewtable where y='yes' group by age order by number desc")
targetage: org.apache.spark.sql.DataFrame = [age: string, number: bigint]

scala> targetage.show()
+-----------+------+
|        age|number|
+-----------+------+
|Middle Aged|  2601|
|      Young|  1539|
|        old|  1131|
|       Teen|    18|
+-----------+------+
```

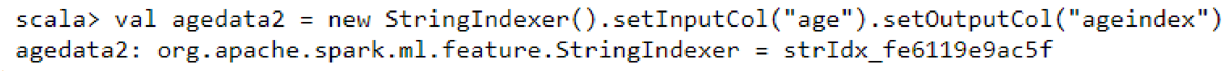
9. The command and its output here assigns generated value of index of the column, by feature engineering

Import of Machine Learning library - StringIndexer



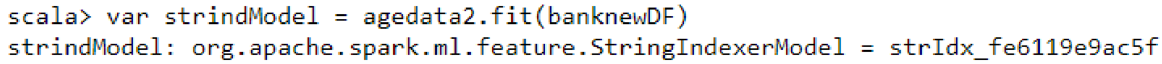
```
scala> import org.apache.spark.ml.feature.StringIndexer
import org.apache.spark.ml.feature.StringIndexer
```

Pipelining with StringIndexer

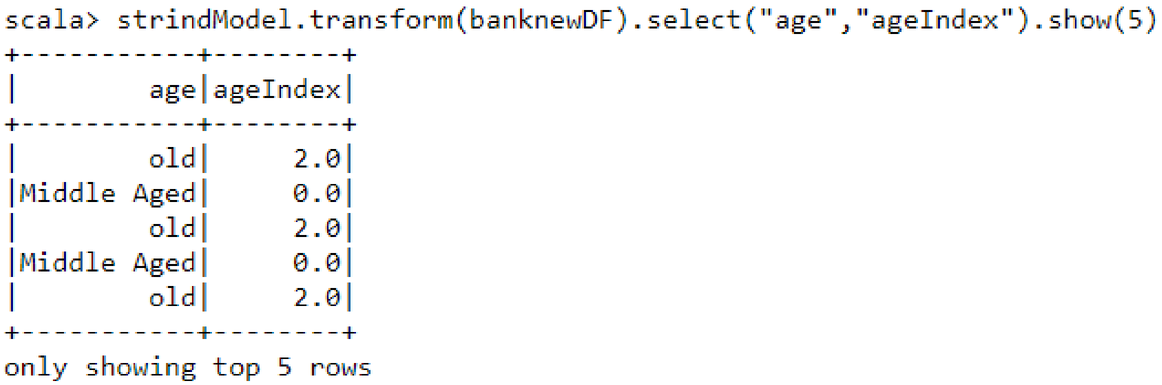


```
scala> val agedata2 = new StringIndexer().setInputCol("age").setOutputCol("ageindex")
agedata2: org.apache.spark.ml.feature.StringIndexer = strIdx_fe6119e9ac5f
```

Fitting the model



```
scala> var strindModel = agedata2.fit(banknewDF)
strindModel: org.apache.spark.ml.feature.StringIndexerModel = strIdx_fe6119e9ac5f
```



```
scala> strindModel.transform(banknewDF).select("age","ageIndex").show(5)
+-----------+--------+
|        age|ageIndex|
+-----------+--------+
|        old|     2.0|
|Middle Aged|     0.0|
|        old|     2.0|
|Middle Aged|     0.0|
|        old|     2.0|
+-----------+--------+
only showing top 5 rows
```

<**END of PROJECT**>