Market Analysis in Banking Domain

Abstract

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.

Presented by: Ankita Agarwal

Background and Objective:

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)

Detailed description of the given dataset:

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')

To Analyze:

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.

Analysis and Interpretations:

1. Load data and create a Spark data frame

Load and Create Spark Data Frame

```
val df = sqlContext.read.format("com.databricks.spark.csv").option("header","true").option("inferSchema","true").option("delimiter",";").load("/FileStore/tables/bank_full-bd3df.csv")

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|
```

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

Marketing Success Rate

```
val suc = df.filter($"y" === "yes").count.toFloat / df.count.toFloat *100
```

```
suc: Float = 11.698481
```

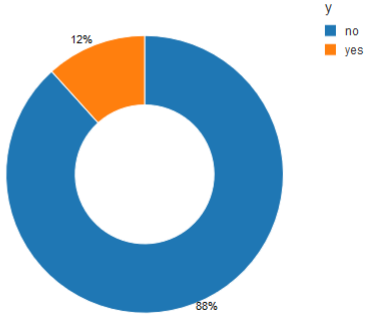
Marketing Fail Rate

```
val fail = df.filter($"y" === "no").count.toFloat / df.count.toFloat *100
```

```
fail: Float = 88.30152
```

Marketing success/failure Rate

```
display(df)
```



3. Give the maximum, mean, and minimum age of the average targeted customer

Max, Min, Min, age of average target customer

```
import org.apache.spark.sql.functions.{min, max, avg}

df.agg(max($"age"),min($"age"), avg($"age")).show()
```

```
+--------+--------+------------------+
|max(age)|min(age)|          avg(age)|
+--------+--------+------------------+
|      95|      18|40.93621021432837|
+--------+--------+------------------+

import org.apache.spark.sql.functions.{min, max, avg}
```

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

Quality of clients by checking average balance, median balance of clients

```
val medBal = sql("SELECT max(balance) as max, min(balance) as min, avg(balance) as average, percentile_approx(balance, 0.5) as median  FROM sample");

medBal.show()
```

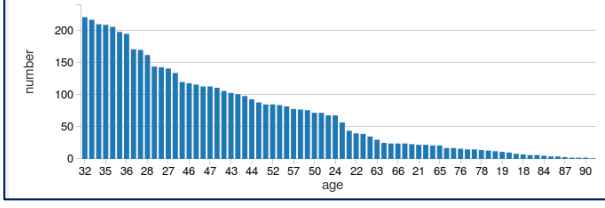
```
+------+-----+------------------+------+
|   max|  min|           average|median|
+------+-----+------------------+------+
|102127|-8019|1362.2720576850766|   448|
+------+-----+------------------+------+

medBal: org.apache.spark.sql.DataFrame = [max: int, min: int ... 2 more fields]
```

5. Check if age matters in marketing subscription for deposit

```
scala> df.groupBy("y").agg(avg($"age")).show
+-----+------------------+
|    y|          avg(age)|
+-----+------------------+
|"yes"|41.670069956513515|
| "no"| 40.83898602274435|
+-----+------------------+
```

```
val age = sqlContext.sql("select age, count(*) as number from bank where y='yes' group by age order by number desc ")
display(age)
```

(1) Spark Jobs



Yes, as can be seen above, age does matter in case of marketing subscription. Most customers who avail to the subscription are in the age range of 32-35years.

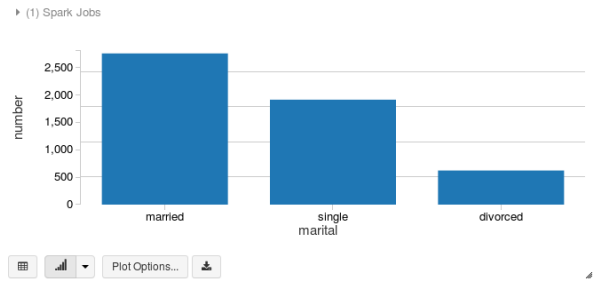
6. Check if marital status mattered for a subscription to deposit

Did marital status mattered for subscription to deposit?

```
df.groupBy($"y".alias("Did the customer Subscribed")).agg(count($"marital").alias("Marital Count")).show
```

```
+--------------------------+-------------+
|Did the customer Subscribed|Marital Count|
+--------------------------+-------------+
|                        no|        39922|
|                       yes|         5289|
+--------------------------+-------------+
```

```
val marital = sqlContext.sql("select marital, count(*) as number from bank where y='yes' group by marital order by number desc ")
display(marital)
```



Command took 1.28 seconds -- by sagarnildass@gmail.com at 4/30/2017, 9:40:30 PM on project

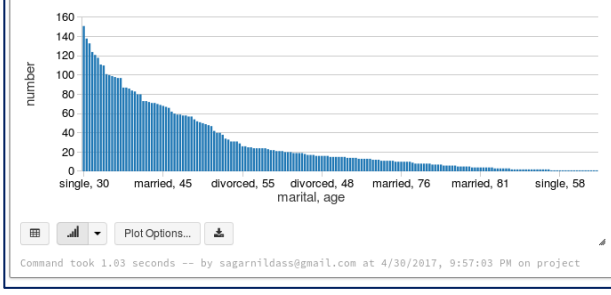
Yes, as can be seen above, marital status also matters in case of marketing subscription. Most customers who avail to the subscription are married.

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

Did age and marital status together mattered for subscription to deposit scheme?

```
df.groupBy("marital","y").count.sort($"count").show
```

```
+--------+---+-----+
| marital|  y|count|
+--------+---+-----+
|divorced|yes|  622|
|  single|yes| 1912|
| married|yes| 2755|
|divorced| no| 4585|
|  single| no|10878|
| married| no|24459|
+--------+---+-----+
```



Command took 1.03 seconds -- by sagarnildass@gmail.com at 4/30/2017, 9:57:03 PM on project

However, if we consider both age and marital status, customers who are single and around 30years mostly avail the subscription.

8. Do feature engineering for the bank and find the right age effect on the campaign.

```
import org.apache.spark.sql.functions.udf

def ageToCategory = udf((age:Int) => {
      age match {
      case t if t < 30 => "young"
      case t if t > 65 => "Old"
      case _ => "mid"
      }
      }
     )

val newdf = df.withColumn("agecat",ageToCategory(df("age"))) // create newcolumn
newdf.groupBy("agecat","y").count().sort($"count".desc).show

+------+---+-----+
|agecat|  y|count|
+------+---+-----+
|   mid| no|35146|
| young| no| 4345|
|   mid|yes| 4041|
| young|yes|  928|
|   Old| no|  431|
|   Old|yes|  320|
+------+---+-----+

import org.apache.spark.sql.functions.udf
ageToCategory: org.apache.spark.sql.expressions.UserDefinedFunction
newdf: org.apache.spark.sql.DataFrame = [age: int, job: string ... 16 more fields]
```

Mostly, middle aged customers avail the marketing subscription.

Programming Codes:

```
import org.apache.spark.sql._
import org.apache.spark.sql.types._
import sqlContext.implicits._
```

Load and Create Spark Data Frame

```
val df =
sqlContext.read.format("com.databricks.spark.csv").option("header","true").option("inferSchema","tr
ue").option("delimiter",";").load("/FileStore/tables/bank_full-bd3df.csv")
df.show()
```

Marketing Success Rate

```
val suc = df.filter($"y" === "yes").count.toFloat / df.count.toFloat *100
```

Marketing Fail Rate

```
val fail = df.filter($"y" === "no").count.toFloat / df.count.toFloat *100
```

Marketing success/failure Rate

```
display(df)
```

Max, Min, Min, age of average target customer

```
import org.apache.spark.sql.functions.{min, max, avg}
df.agg(max($"age"),min($"age"), avg($"age")).show()
```

Quality of clients by checking average balance, median balance of clients

```
val medBal = sql("SELECT max(balance) as max, min(balance) as min, avg(balance) as average,
percentile_approx(balance, 0.5) as median  FROM sample");
medBal.show()
```

Did age mattered for subscription to deposit?

```
df.groupBy("y").agg(avg($"age")).show

val age = sqlContext.sql("select age, count(*) as number from bank where y='yes' group by age order
by
number desc ").show()
```

Did marital status mattered for subscription to deposit?

```
df.groupBy($"y".alias("Did the customer Subscribed")).agg(count($"marital").alias("Marital Count")).show

val marital = sqlContext.sql("select marital, count(*) as number from bank where y='yes' group by marital order by number desc ").show()
```

Did age and marital status together mattered for subscription to deposit scheme?

```
df.groupBy("marital","y").count.sort($"count").show

val age_marital = sqlContext.sql("select age, marital, count(*) as number from bank where y='yes' group
by age,marital order by number desc ").show()
```

Feature engineering for age column and find right age effect on campaign

```
import org.apache.spark.sql.functions.udf

def ageToCategory = udf((age:Int) => {
    age match {
    case t if t < 30 => "young"
    case t if t > 65 => "Old"
    case _ => "mid"
    }
    }
   )

val newdf = df.withColumn("agecat",ageToCategory(df("age"))) // create newcolumn
newdf.groupBy("agecat","y").count().sort($"count".desc).show
```

----------The End----------