Project_3_Market_Analysis_in_Banking_Domain

January 16, 2021

1 Project Title: Market Analysis in Banking Domain

1.1 By: Kuruma Manirathnam Babu

1.1.1 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.

Analysis tasks to be done-:

1. Load data and create a Spark data frame.

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

Give marketing failure rate.

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

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

3. Check if age matters in marketing subscription for deposit.

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

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

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

[1]:
```
# lets import the required libraries
import pyspark
```

[2]:
```
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
sc = SparkContext.getOrCreate();
spark= SparkSession(sc)
```

[3]:
```
import pyspark.sql.functions as f
from pyspark.sql.functions import lit, when, col, regexp_extract, desc
from pyspark.sql import SQLContext
from pyspark.sql import DataFrameStatFunctions as statFunc
from pyspark.sql.functions import explode, col, udf, mean as mean, stddev as␣
 ↪stddev
from pyspark.sql.types import IntegerType,StringType
from pyspark.sql.functions import udf
```

[4]:
```
from pyspark.sql import *
from pyspark.sql.types import *
```

1. Load data and create a Spark data frame:

[5]:
```
bank_df= spark.read.csv("./dataset/Marketing_Analysis.
 ↪csv",inferSchema=True,header=True)
```

[6]:
```
# lets see the dataframe structure
bank_df
```

[6]:
```
DataFrame[age: int, job: string, marital: string, education: string, default:
 string, balance: int, housing: string, loan: string, contact: string, day: int,
 month: string, duration: int, campaign: int, pdays: int, previous: int,
 poutcome: string, y: string]
```

[7]:
```
# lets see the records
bank_df.show()
```

```
+---+-----------+-------+---------+-------+-------+-------+----+-------+---+--
---+--------+--------+-----+--------+--------+---+
|age|        job| marital|education|default|balance|housing|loan|contact|day|mo
nth|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
```

[8]:
```
# lets see the schema
bank_df.printSchema()
```

```
root
 |-- age: integer (nullable = true)
 |-- job: string (nullable = true)
 |-- marital: string (nullable = true)
```

```
|-- education: string (nullable = true)
|-- default: string (nullable = true)
|-- balance: integer (nullable = true)
|-- housing: string (nullable = true)
|-- loan: string (nullable = true)
|-- contact: string (nullable = true)
|-- day: integer (nullable = true)
|-- month: string (nullable = true)
|-- duration: integer (nullable = true)
|-- campaign: integer (nullable = true)
|-- pdays: integer (nullable = true)
|-- previous: integer (nullable = true)
|-- poutcome: string (nullable = true)
|-- y: string (nullable = true)
```

[9]:
```
# lets check the count of records
records_count= bank_df.count()
print('Total no. of records:',records_count)
```

```
Total no. of records: 45211
```

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

Give marketing failure rate

[10]:
```
# To know the success and failure rate,
# First we need to know the no. of people who are subscribed and who are not.
# lets find out the people who are subscribed:
people_subscribed= bank_df.filter(bank_df.y=='yes').count()
print('No. of people who are subscribed:',people_subscribed)
# now lets find the people who are not subscribed:
people_not_subscribed= bank_df.filter(bank_df.y=='no').count()
print('No. of people who are not subscribed:',people_not_subscribed)
```

```
No. of people who are subscribed: 5289
No. of people who are not subscribed: 39922
```

[11]:
```
# lets find out marketing success rate:
success_rate= (people_subscribed/records_count)*100
print('Marketing Success Rate:',success_rate)
```

```
Marketing Success Rate: 11.698480458295547
```

[12]:
```
# lets find out marketing failure rate:
failure_rate= (people_not_subscribed/records_count)*100
print('Marketing Failure Rate:',failure_rate)
```

```
Marketing Failure Rate: 88.30151954170445
```

2.b) Give the maximum, mean, and minimum age of the average targeted customer

[13]:
```
# maximum age of the average targeted customer
max_age= bank_df.agg({'age':'max'}).show()
```

```
+--------+
|max(age)|
+--------+
|      95|
+--------+
```

[14]:
```
# mean age of the average targeted customer
mean_age= bank_df.agg({'age':'mean'}).show()
```

```
+-----------------+
|         avg(age)|
+-----------------+
|40.93621021432837|
+-----------------+
```

[15]:
```
# minimum age of the average targeted customer
min_age= bank_df.agg({'age':'min'}).show()
```

```
+--------+
|min(age)|
+--------+
|      18|
+--------+
```

2.c) Check the quality of customers by checking average balance, median balance of customers

[16]:
```
# average balance of customers
avg_balance= bank_df.agg({'balance':'mean'}).show()
```

```
+------------------+
|      avg(balance)|
+------------------+
|1362.2720576850766|
+------------------+
```

[17]:
```
# median balance of customers
median_balance= bank_df.approxQuantile('balance',[0.5],0)
print('The Median of Balance is:',median_balance)
```

```
The Median of Balance is: [448.0]
```

3. Check if age matters in marketing subscription for deposit

[18]:
```
# lets first check the age group with 'yes' and 'no' records
bank_df.groupBy('age').pivot('y').count().show()
```

```
+---+----+----+
|age|  no| yes|
+---+----+----+
| 31|1790| 206|
| 85|   1|   4|
| 65|  38|  21|
| 53| 806|  85|
| 78|  16|  14|
| 34|1732| 198|
| 81|  11|   6|
| 28| 876| 162|
| 76|  16|  16|
| 27| 768| 141|
| 26| 671| 134|
| 44|1043|  93|
| 22|  89|  40|
| 93|null|   2|
| 47| 975| 113|
| 52| 826|  85|
| 86|   5|   4|
| 20|  35|  15|
| 40|1239| 116|
| 94|   1|null|
+---+----+----+
only showing top 20 rows
```

[19]:
```
# age matters in marketing subscription
# sorting the subscribed people group by age in descending order
bank_df.where(bank_df.y=='yes').groupBy(bank_df.age).count().
 ↪sort(desc("count")).show()
```

```
+---+-----+
|age|count|
+---+-----+
| 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|
| 47|  113|
| 25|  113|
| 42|  111|
+---+-----+
only showing top 20 rows
```

Conclusion: Yes, age matters and We can say that the age group of 30 subscribed the most.

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

[20]:
```
#lets first check the subscription scenario in terms of age and marital status
bank_df.groupBy('age','y').pivot('marital').agg(f.count('y')).show()
```

```
+---+---+--------+-------+------+
|age|  y|divorced|married|single|
+---+---+--------+-------+------+
| 78| no|       6|     10|  null|
| 20| no|    null|      2|    33|
| 56|yes|      13|     49|     6|
| 28|yes|       4|     20|   138|
| 29|yes|       5|     33|   133|
| 86|yes|       1|      2|     1|
| 71| no|       3|     25|     1|
| 57| no|     133|    584|    33|
| 79|yes|       2|      8|  null|
| 22|yes|    null|   null|    40|
| 31|yes|      15|     80|   111|
| 42| no|     165|    770|   196|
| 87|yes|       1|      2|  null|
| 59|yes|      16|     66|     6|
```

```
| 34|yes|      11|    118|    69|
| 25| no|       6|     84|   324|
| 63| no|       3|     43|     1|
| 23|yes|    null|      2|    42|
| 24| no|       1|     43|   190|
| 64| no|       5|     34|  null|
+---+---+--------+-------+------+
only showing top 20 rows
```

[21]:
```
# marital status mattered for a subscription
bank_df.groupBy('marital').pivot('y').count().show()
```

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

Conclusion: Since most of the subscribers are married. we can say that marital status matters a lot for subscription.

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

[22]:
```
# age and marital status of people who subscribed
bank_df.where(bank_df.y=='yes').groupBy(bank_df.age).pivot('marital').agg(f.
 ↪count('y')).show()
```

```
+---+--------+-------+------+
|age|divorced|married|single|
+---+--------+-------+------+
| 31|      15|     80|   111|
| 85|       1|      3|  null|
| 65|       2|     19|  null|
| 53|      18|     60|     7|
| 78|       6|      8|  null|
| 34|      11|    118|    69|
| 81|       2|      4|  null|
| 28|       4|     20|   138|
| 76|       6|     10|  null|
| 27|       2|     29|   110|
| 26|    null|     13|   121|
| 44|      21|     48|    24|
| 22|    null|   null|    40|
```

```
| 93|    null|      2|  null|
| 47|      10|     83|    20|
| 52|      10|     67|     8|
| 86|       1|      2|     1|
| 40|      12|     73|    31|
| 20|    null|      1|    14|
| 57|      15|     58|     5|
+---+--------+-------+------+
only showing top 20 rows
```

[23]:
```
# age and marital status of people who are not subscribed
bank_df.where(bank_df.y=='no').groupBy(bank_df.age).pivot('marital').agg(f.
 ↪count('y')).show()
```

```
+---+--------+-------+------+
|age|divorced|married|single|
+---+--------+-------+------+
| 31|      83|    801|   906|
| 85|    null|      1|  null|
| 65|       7|     31|  null|
| 53|     145|    597|    64|
| 78|       6|     10|  null|
| 34|     138|   1013|   581|
| 81|       6|      5|  null|
| 28|      12|    305|   559|
| 76|       2|     14|  null|
| 26|      20|    157|   494|
| 27|      16|    204|   548|
| 44|     163|    734|   146|
| 22|    null|      9|    80|
| 47|     152|    743|    80|
| 52|     140|    632|    54|
| 86|       1|      4|  null|
| 40|     157|    856|   226|
| 20|    null|      2|    33|
| 94|       1|   null|  null|
| 57|     133|    584|    33|
+---+--------+-------+------+
only showing top 20 rows
```

Conclusion: As we can see age group of 30 are mostly subscribed irrespective of marital status.

Hence, we can say age and marital status together has minimum effect on subscription.

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

[24]:
```
#lets first find out the age group and subscribers count in descending order
bank_df.where(bank_df.y=='yes').groupBy(bank_df.age).count().
 ↪sort(desc('count')).show()
```

```
+---+-----+
|age|count|
+---+-----+
| 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|
| 47|  113|
| 25|  113|
| 42|  111|
+---+-----+
only showing top 20 rows
```

[25]:
```
#lets create age type column based on the age range.
e= bank_df.withColumn("AgeType",f.when(((bank_df.age>=15) &(bank_df.age<=30)),␣
 ↪'YOUNG').when(((bank_df.age>=31) &(bank_df.age<=59)), 'MID').when(bank_df.
 ↪age>=60, 'OLD'))
e.show()
```

```
+---+-----------+-------+---------+-------+-------+-------+----+-------+---+--
---+--------+--------+-----+--------+--------+---+-------+
|age|        job| marital|education|default|balance|housing|loan|contact|day|mo
nth|duration|campaign|pdays|previous|poutcome|  y|AgeType|
+---+-----------+-------+---------+-------+-------+-------+----+-------+---+--
---+--------+--------+-----+--------+--------+---+-------+
| 58|  management| married| tertiary|     no|   2143|    yes|  no|unknown|  5|
may|     261|       1|   -1|       0| unknown| no|    MID|
| 44|  technician|  single|secondary|     no|     29|    yes|  no|unknown|  5|
```

```
may|     151|       1|   -1|       0| unknown| no|    MID|
| 33|entrepreneur| married|secondary|     no|      2|    yes| yes|unknown|  5|
may|      76|       1|   -1|       0| unknown| no|    MID|
| 47| blue-collar| married|  unknown|     no|   1506|    yes|  no|unknown|  5|
may|      92|       1|   -1|       0| unknown| no|    MID|
| 33|     unknown|  single|  unknown|     no|      1|     no|  no|unknown|  5|
may|     198|       1|   -1|       0| unknown| no|    MID|
| 35|  management| married| tertiary|     no|    231|    yes|  no|unknown|  5|
may|     139|       1|   -1|       0| unknown| no|    MID|
| 28|  management|  single| tertiary|     no|    447|    yes| yes|unknown|  5|
may|     217|       1|   -1|       0| unknown| no|  YOUNG|
| 42|entrepreneur|divorced| tertiary|    yes|      2|    yes|  no|unknown|  5|
may|     380|       1|   -1|       0| unknown| no|    MID|
| 58|     retired| married|  primary|     no|    121|    yes|  no|unknown|  5|
may|      50|       1|   -1|       0| unknown| no|    MID|
| 43|  technician|  single|secondary|     no|    593|    yes|  no|unknown|  5|
may|      55|       1|   -1|       0| unknown| no|    MID|
| 41|      admin.|divorced|secondary|     no|    270|    yes|  no|unknown|  5|
may|     222|       1|   -1|       0| unknown| no|    MID|
| 29|      admin.|  single|secondary|     no|    390|    yes|  no|unknown|  5|
may|     137|       1|   -1|       0| unknown| no|  YOUNG|
| 53|  technician| married|secondary|     no|      6|    yes|  no|unknown|  5|
may|     517|       1|   -1|       0| unknown| no|    MID|
| 58|  technician| married|  unknown|     no|     71|    yes|  no|unknown|  5|
may|      71|       1|   -1|       0| unknown| no|    MID|
| 57|    services| married|secondary|     no|    162|    yes|  no|unknown|  5|
may|     174|       1|   -1|       0| unknown| no|    MID|
| 51|     retired| married|  primary|     no|    229|    yes|  no|unknown|  5|
may|     353|       1|   -1|       0| unknown| no|    MID|
| 45|      admin.|  single|  unknown|     no|     13|    yes|  no|unknown|  5|
may|      98|       1|   -1|       0| unknown| no|    MID|
| 57| blue-collar| married|  primary|     no|     52|    yes|  no|unknown|  5|
may|      38|       1|   -1|       0| unknown| no|    MID|
| 60|     retired| married|  primary|     no|     60|    yes|  no|unknown|  5|
may|     219|       1|   -1|       0| unknown| no|    OLD|
| 33|    services| married|secondary|     no|      0|    yes|  no|unknown|  5|
may|      54|       1|   -1|       0| unknown| no|    MID|
+---+------------+--------+---------+-------+-------+-------+----+-------+---+--
---+--------+--------+-----+--------+--------+---+-------+
only showing top 20 rows
```

[26]:
```
#lets see the age type column of this new df that we just created
e.select('AgeType','y').show()
```

```
+-------+---+
|AgeType|  y|
+-------+---+
```

```
|    MID| no|
|    MID| no|
|    MID| no|
|    MID| no|
|    MID| no|
|    MID| no|
|  YOUNG| no|
|    MID| no|
|    MID| no|
|    MID| no|
|    MID| no|
|  YOUNG| no|
|    MID| no|
|    MID| no|
|    MID| no|
|    MID| no|
|    MID| no|
|    MID| no|
|    OLD| no|
|    MID| no|
+-------+---+
only showing top 20 rows
```

[27]:
```
#lets find out the right age effect on the campaign
e.groupBy('AgeType').pivot('y').count().show()
```

```
+-------+-----+----+
|AgeType|   no| yes|
+-------+-----+----+
|    MID|32853|3544|
|  YOUNG| 5885|1145|
|    OLD| 1184| 600|
+-------+-----+----+
```

Conclusion: Middle Age Group has the more effect on the campaign.