MARKET ANALYSIS IN BANKING DOMAIN

Ashish Rai

Setup

```
scala>  val bankRDD = sc.textFile("./Data/bank_original.csv").
     |  map(_.replace("\"", "").
     |  replace(";", ",")
     |  )
bankRDD: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at map at <console>:24

scala>  bankRDD.coalesce(1).saveAsTextFile("./Data/BankClean")

scala>  val bankDF = spark.read.
     |  option("header",true).
     |  option("inferSchema",true).
     |  csv("./data/BankClean/part-00000")
bankDF: org.apache.spark.sql.DataFrame = [age: int, job: string ... 15 more fields]

scala> bankDF.show(5)
+---+-----------+-------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+-----+--------+--------+---+
|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|
+---+-----------+-------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+-----+--------+--------+---+
only showing top 5 rows


scala> bankDF.count
res2: Long = 45211

scala>  bankDF.createOrReplaceTempView("bank")
```

Campaign Performance

- Success Rate **11.7%**
- Failure Rate **88.3%**

```
scala>  spark.sql("""
     |  SELECT
     |  ROUND((subscriber.count/all.count)*100, 1) AS success_rate,
     |  ROUND((nonsubscriber.count/all.count)*100, 1) AS failure_rate
     |  FROM
     |  (SELECT COUNT(*) AS count FROM bank WHERE y='yes') subscriber,
     |  (SELECT COUNT(*) AS count FROM bank WHERE y='no') nonsubscriber,
     |  (SELECT COUNT(*) AS count FROM bank) all
     |  """).show
+------------+------------+
|success_rate|failure_rate|
+------------+------------+
|        11.7|        88.3|
+------------+------------+
```

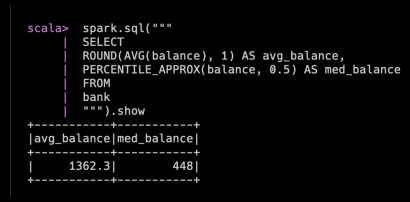
Customers Age Statistics

- Minimum 18
- Maximum 95
- Average 40.94

```
scala>  spark.sql("""
     |  SELECT
     |  MIN(age) AS min_age,
     |  MAX(age) AS max_age,
     |  ROUND(AVG(age), 1) AS avg_age
     |  FROM
     |  bank
     |  """).show
+-------+-------+-------+
|min_age|max_age|avg_age|
+-------+-------+-------+
|     18|     95|   40.9|
+-------+-------+-------+
```

Customer Quality Check (Balance)

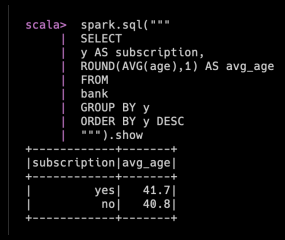
- Due to skewness, we can conclude that most of the customers belong to **lower** than average income strata.



```
scala>  spark.sql("""
     |  SELECT
     |  ROUND(AVG(balance), 1) AS avg_balance,
     |  PERCENTILE_APPROX(balance, 0.5) AS med_balance
     |  FROM
     |  bank
     |  """).show
+-----------+-----------+
|avg_balance|med_balance|
+-----------+-----------+
|     1362.3|        448|
+-----------+-----------+
```

Does Age matter?

- Due to significantly low difference between average age of subscribers vs non-subscribers, we can conclude that age is **not significant** factor for campaign success.



```
scala>  spark.sql("""
     |  SELECT
     |  y AS subscription,
     |  ROUND(AVG(age),1) AS avg_age
     |  FROM
     |  bank
     |  GROUP BY y
     |  ORDER BY y DESC
     |  """).show
+------------+-------+
|subscription|avg_age|
+------------+-------+
|         yes|   41.7|
|          no|   40.8|
+------------+-------+
```

Does Marital status matter?

- **Single** customers have higher average chance to respond positively, concluding that marital status does matter.

```
scala>  spark.sql("""
     |  SELECT
     |  ROUND((allsubscribers.count/all.count)*100,2) AS overall_success_rate,
     |  ROUND((divorcedsubscribers.count/alldivorced.count)*100,2) AS divorced_success_rate,
     |  ROUND((marriedsubscribers.count/allmarried.count)*100,2) AS married_success_rate,
     |  ROUND((singlesubscribers.count/allsingle.count)*100,2) AS single_success_rate
     |  FROM
     |  (SELECT COUNT(*) AS count FROM bank WHERE y='yes') allsubscribers,
     |  (SELECT COUNT(*) AS count FROM bank) all,
     |  (SELECT COUNT(*) AS count FROM bank WHERE marital='divorced' AND y='yes') divorcedsubscribers,
     |  (SELECT COUNT(*) AS count FROM bank WHERE marital='divorced') alldivorced,
     |  (SELECT COUNT(*) AS count FROM bank WHERE marital='married' AND y='yes') marriedsubscribers,
     |  (SELECT COUNT(*) AS count FROM bank WHERE marital='married') allmarried,
     |  (SELECT COUNT(*) AS count FROM bank WHERE marital='single' AND y='yes') singlesubscribers,
     |  (SELECT COUNT(*) AS count FROM bank WHERE marital='single') allsingle
     |  """).show
+-------------------+---------------------+-------------------+-------------------+
|overall_success_rate|divorced_success_rate|married_success_rate|single_success_rate|
+-------------------+---------------------+-------------------+-------------------+
|               11.7|                11.95|              10.12|              14.95|
+-------------------+---------------------+-------------------+-------------------+
```

Does Age + Marital status matter?

- **(Old + Married or Divorced)** customers give significantly higher positive reponse to subscription deposit scheme.

```
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE age_group='middle_age') allmarried,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE y='yes' AND age_group='old') oldsubscribers,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE age_group='old') allold""").show
+----------+---------+-----+
|youngadult|middlеage|  old|
+----------+---------+-----+
|     12.16|     9.21|33.63|
+----------+---------+-----+


scala>  spark.sql("""SELECT
     |  ROUND((youngadultmarriedsubscribers.count/allyoungadultmarried.count)*100, 2) AS youngadult_married,
     |  ROUND((middleagemarriedsubscribers.count/allmiddleagemarried.count)*100, 2) AS middleage_married,
     |  ROUND((oldmarriedsubscribers.count/alloldmarried.count)*100, 2) AS old_married FROM
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='married' AND y='yes' AND age_group='young_adult') youngadultmarriedsubscribers,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='married' AND age_group='young_adult') allyoungadultmarried,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='married' AND y='yes' AND age_group='middle_age') middleagemarriedsubscribers,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='married' AND age_group='middle_age') allmiddleagemarried,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='married' AND y='yes' AND age_group='old') oldmarriedsubscribers,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='married' AND age_group='old') alloldmarried""").show
+-----------------+----------------+-----------+
|youngadult_married|middleage_married|old_married|
+-----------------+----------------+-----------+
|             9.21|            8.59|      33.03|
+-----------------+----------------+-----------+


scala>  spark.sql("""SELECT
     |  ROUND((youngadultdivorcedsubscribers.count/allyoungadultdivorced.count)*100,2) AS youngadult_divorced,
     |  ROUND((middleagedivorcedsubscribers.count/allmiddleagedivorced.count)*100,2) AS middleage_divorced,
     |  ROUND((olddivorcedsubscribers.count/allolddivorced.count)*100,2) AS oldage_divorced FROM
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='divorced' AND age_group='young_adult' AND y='yes') youngadultdivorcedsubscribers,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='divorced' AND age_group='young_adult') allyoungadultdivorced,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='divorced' AND age_group='middle_age' AND y='yes') middleagedivorcedsubscribers,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='divorced' AND age_group='middle_age') allmiddleagedivorced,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='divorced' AND age_group='old' AND y='yes') olddivorcedsubscribers,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='divorced' AND age_group='old') allolddivorced""").show
+-----------------+----------------+---------------+
|youngadult_divorced|middleage_divorced|oldage_divorced|
+-----------------+----------------+---------------+
|            10.56|           10.14|          38.94|
+-----------------+----------------+---------------+


scala>  spark.sql("""SELECT
     |  ROUND((youngadultsinglesubscribers.count/allyoungadultsingle.count)*100,2) AS youngadult_single,
     |  ROUND((middleagesinglesubscribers.count/allmiddleagesingle.count)*100,2) AS middleage_single,
     |  ROUND((oldsinglesubscribers.count/alloldsingle.count)*100,2) AS oldage_single FROM
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='single' AND age_group='young_adult' AND y='yes') youngadultsinglesubscribers,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='single' AND age_group='young_adult') allyoungadultsingle,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='single' AND age_group='middle_age' AND y='yes') middleagesinglesubscribers,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='single' AND age_group='middle_age') allmiddleagesingle,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='single' AND age_group='old' AND y='yes') oldsinglesubscribers,
     |  (SELECT COUNT(*) AS count FROM bankAgeGroupMarital WHERE marital='single' AND age_group='old') alloldsingle
     |  """).show
+----------------+---------------+-------------+
|youngadult_single|middleage_single|oldage_single|
+----------------+---------------+-------------+
|           15.56|          11.88|        21.31|
+----------------+---------------+-------------+
```

Age for target marketing

- Old age customers i.e **60+ years**

```
scala> val minAge = bankDF.select(min($"age")).collect()(0)(0).asInstanceOf[Int]
minAge: Int = 18

scala>

scala>  val maxAge = bankDF.select(max($"age")).collect()(0)(0).asInstanceOf[Int]
maxAge: Int = 95

scala>

scala>  var age = minAge
age: Int = 18

scala>

scala> scala.math.BigDecimal
res94: math.BigDecimal.type = scala.math.BigDecimal$@4b7191cf

scala>

scala>  while (age < maxAge) {
     |     println(age + " - " + (age + 5) + " -> " +   BigDecimal(bankDF.where($"age" >= age and $"age" < age+5 and $"y" === "yes").co
unt / bankDF.where($"age" >= age and $"age" < age+5).count.toFloat*100).setScale(2, BigDecimal.RoundingMode.HALF_UP).toDouble)
     |
     |     age = age + 5
     |  }
warning: one deprecation (since 2.11.0); for details, enable `:setting -deprecation' or `:replay -deprecation'
18 - 23 -> 31.15
23 - 28 -> 18.21
28 - 33 -> 12.12
33 - 38 -> 10.56
38 - 43 -> 9.27
43 - 48 -> 9.23
48 - 53 -> 8.73
53 - 58 -> 9.5
58 - 63 -> 15.17
63 - 68 -> 40.67
68 - 73 -> 41.11
73 - 78 -> 45.92
78 - 83 -> 40.98
83 - 88 -> 44.9
88 - 93 -> 44.44
93 - 98 -> 60.0
```

END