Use Case 3

Gabriel D Hofer

July 15, 2021

1. Load data and create spark data frame

```
scala> val df = spark.read.format("csv").option("header", "true").load("/home
    /gabriel/Data/bank-full.csv")

scala> df.createOrReplaceTempView("df")
```

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

```
scala> df.agg((sum(when(col("y") === lit("yes"),1).otherwise(0))/count("*")).
     | as("marketing_success_rate")).
     | show()
+----------------------+
|marketing_success_rate|
+----------------------+
|   0.11698480458295547|
+----------------------+
```

3. Check max, min, Mean and median age of average targeted customer

```
scala> spark.sql("SELECT MAX(age) AS max_age, MIN(age) AS min_age, AVG(age)
    AS avg_age, percentile_approx(age,0.5) AS median_age FROM df").show()
+-------+-------+-----------------+----------+
|max_age|min_age|          avg_age|median_age|
+-------+-------+-----------------+----------+
|     95|     18|40.93621021432837|      39.0|
+-------+-------+-----------------+----------+
```

4. Check quality of clients by checking average balance, median balance of clients

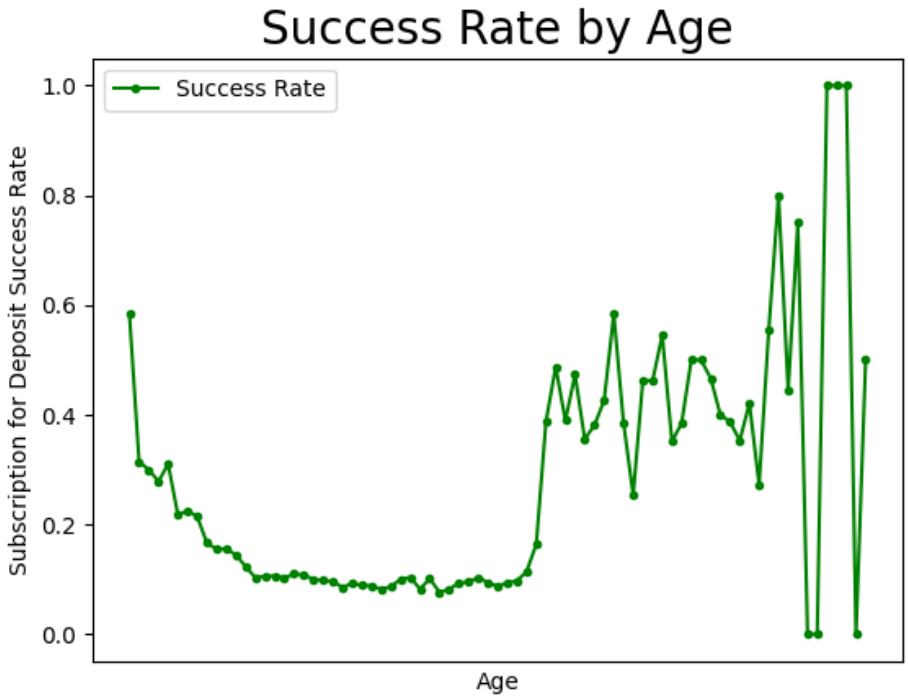
```
scala> spark.sql("SELECT AVG(balance) AS average_balance, percentile_approx(
    balance, 0.5) AS median_balance FROM df").show()
+------------------+--------------+
|   average_balance|median_balance|
+------------------+--------------+
|1362.2720576850766|         448.0|
+------------------+--------------+
```

5. Check if age matters in marketing subscription for deposit

```
scala> df.
     | groupBy(col("age")).
```

```
     | agg((sum(when(col("y") === lit("yes"),1).otherwise(0))/count("*")).
     | as("subscription_for_deposit_success_rate")).
     | orderBy(asc("age")).
     | show(100)
+---+-------------------------------------+
|age|subscription_for_deposit_success_rate|
+---+-------------------------------------+
| 18|                   0.5833333333333334|
| 19|                    0.3142857142857143|
| 20|                                  0.3|
| 21|                   0.27848101265822783|
| 22|                   0.31007751937984496|
| 23|                   0.21782178217821782|
| 24|                    0.2251655629139073|
| 25|                    0.2144212523719165|
| 26|                   0.16645962732919253|
| 27|                    0.1551155115511551|
| 28|                   0.15606936416184972|
| 29|                   0.14430379746835442|
| 30|                   0.12350597609561753|
| 31|                   0.10320641282565131|
| 32|                   0.10599520383693045|
| 33|                   0.10649087221095335|
| 34|                   0.10259067357512953|
| 35|                   0.11034846884899684|
| 36|                    0.1079734219269103|
| 37|                   0.10023584905660378|
| 38|                   0.09822646657571624|
| 39|                   0.09616677874915938|
| 40|                   0.08560885608856089|
| 41|                   0.09295120061967467|
| 42|                    0.0893719806763285|
| 43|                   0.08871662360034453|
| 44|                    0.0818661971830986|
| 45|                   0.08717105263157894|
| 46|                   0.10042553191489362|
| 47|                   0.10386029411764706|
| 48|                   0.08224674022066199|
| 49|                   0.10160965794768612|
| 50|                   0.07667731629392971|
| 51|                   0.08226495726495726|
| 52|                   0.09330406147091108|
| 53|                   0.09539842873176206|
| 54|                   0.10357583230579531|
| 55|                   0.09429280397022333|
| 56|                   0.08740359897172237|
| 57|                   0.09420289855072464|
| 58|                    0.0972972972972973|
| 59|                   0.11428571428571428|
| 60|                    0.1644295302013423|
| 61|                    0.3877551020408163|
| 62|                                0.4875|
| 63|                   0.38961038961038963|
| 64|                   0.47297297297297297|
| 65|                    0.3559322033898305|
| 66|                   0.38095238095238093|
| 67|                   0.42592592592592593|
| 68|                   0.5833333333333334|
| 69|                   0.38636363636363635|
| 70|                    0.2537313432835821|
```

```
| 71|                   0.46296296296296297|
| 72|                   0.46153846153846156|
| 73|                    0.5454545454545454|
| 74|                   0.35135135135135137|
| 75|                   0.38461538461538464|
| 76|                                  0.5|
| 77|                                  0.5|
| 78|                    0.4666666666666667|
| 79|                                  0.4|
| 80|                    0.3870967741935484|
| 81|                   0.35294117647058826|
| 82|                   0.42105263157894735|
| 83|                    0.2727272727272727|
| 84|                    0.5555555555555556|
| 85|                                  0.8|
| 86|                    0.4444444444444444|
| 87|                                 0.75|
| 88|                                  0.0|
| 89|                                  0.0|
| 90|                                  1.0|
| 92|                                  1.0|
| 93|                                  1.0|
| 94|                                  0.0|
| 95|                                  0.5|
+---+-------------------------------------+
```



As seen from the data and the plot, the success rate increased significantly after age 61. The success rate is also higher for people in the age range 18-29.

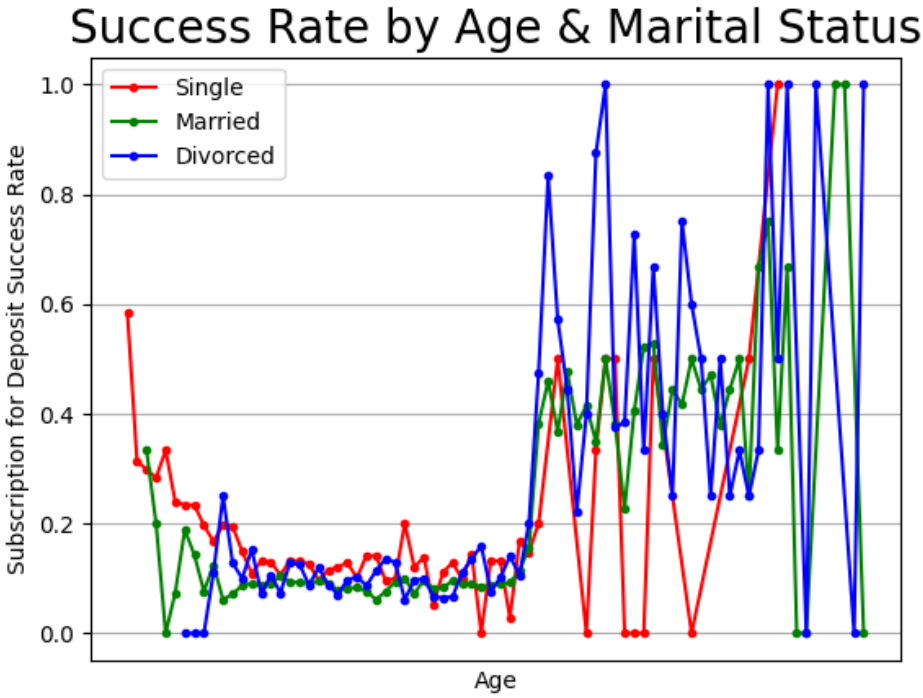
6. Check if marital status mattered for subscription to deposit.

```
scala> df.
     | groupBy(col("marital")).
     | agg((sum(when(col("y") === lit("yes"),1).otherwise(0))/count("*")).
     | as("subscription_for_deposit_success_rate")).
     | show(10)
+--------+-------------------------------------+
| marital|subscription_for_deposit_success_rate|
+--------+-------------------------------------+
|divorced|                  0.11945458037257538|
| married|                  0.10123465863158668|
|  single|                   0.1494917904612979|
+--------+-------------------------------------+
```

It appears that the Single category had a higher subscription success rate.

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

```
scala> df.
     | groupBy(col("age"),col("marital")).
     | agg((sum(when(col("y") === lit("yes"),1).otherwise(0))/count("*")).
     | as("subscription_for_deposit_success_rate")).
     | show()
+---+--------+-------------------------------------+
|age| marital|subscription_for_deposit_success_rate|
+---+--------+-------------------------------------+
| 73|divorced|                   0.6666666666666666|
| 59|divorced|                  0.10596026490066225|
| 21|  single|                  0.28378378378378377|
| 53|divorced|                  0.11042944785276074|
| 69|divorced|                                0.375|
| 18|  single|                   0.5833333333333334|
| 29| married|                  0.07127429805615551|
| 67|  single|                   0.3333333333333333|
| 27|divorced|                   0.1111111111111111|
| 58| married|                  0.09246575342465753|
| 46|divorced|                  0.12953367875647667|
| 80| married|                   0.3793103448275862|
| 70|  single|                                  0.0|
| 54|divorced|                  0.13559322033898305|
| 66| married|                  0.41509433962264153|
| 68|  single|                                  0.5|
| 56| married|                  0.08687943262411348|
| 70| married|                  0.22641509433962265|
| 72|  single|                                  0.0|
| 33|  single|                  0.13002680965147453|
+---+--------+-------------------------------------+
only showing top 20 rows
```



As seen from the data and the plot, we can conclude that age and marital status both mattered for subscription to deposit scheme.