


```

success: Unit = ()

scala> val failure = sqlContext.sql("select (a.not_subscribed/b.total)*100 as failure_percent from (select count(*) as not_subscribed from bank where y='no') a,(select count(*) as total from bank) b").show()
+-----+
| failure_percent |
+-----+
| 88.30151954170445 |
+-----+

failure: Unit = ()

scala>

```

```

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

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

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

```

```
scala> df.select("age").summary().show()
-----+-----+
summary|      age|
-----+-----+
count  |    45211|
mean   |40.93621021432837|
stddev |10.61876204097542|
min    |      18|
25%    |      33|
50%    |      39|
75%    |      48|
max    |      95|
-----+-----+
```

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

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

median: Unit = ()

scala> █
```

```
scala> 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|
+-----+-----+

scala> 
```

```
scala> val ageRDD = sqlContext.udf.register("ageRDD", (age: Int) => {
  | if (age < 20)
  |   "Teen"
  | else if (age > 20 && age <= 32)
  |   "Young"
  | else if (age > 33 && age <= 55)
  |   "Middle Aged"
  | else
  |   "Old"
  | })
ageRDD: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function1>,StringType,Some(List(IntegerType)))
```

```
scala> val age_target = sqlContext.sql("select age, count(*) as number from bank_new where y='yes' group by age order by number desc ").show()
+-----+-----+
|age|number|
+-----+-----+
|Middle Aged|2601|
|Young|1539|
|Old|1131|
|Teen|18|
+-----+-----+

age_target: Unit = ()
```

```
scala>

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
```

```
scala>

scala> strIndModel.transform(banknewDF).select("age","ageIndex").show(10)
+-----+-----+
|      age|ageIndex|
+-----+-----+
|      Old|      2.0|
|Middle Aged|      0.0|
|      Old|      2.0|
|Middle Aged|      0.0|
|      Old|      2.0|
|Middle Aged|      0.0|
|      Young|      1.0|
|Middle Aged|      0.0|
|      Old|      2.0|
|Middle Aged|      0.0|
+-----+-----+
only showing top 10 rows

scala>
```

So we can conclude from the Feature Engineering that It is the ‘Middle Aged’ people between age 33 and 55 who should be the targeted customers as they subscribe the most.