Hadoop Bigdata Project

**Market Analysis in Banking Domain**

**Domain**: Banking (Market Analysis)

**Background and Objective:**

Your client, 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-:**

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

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

* **Importing spark sql datatypes**

import org.apache.spark.sql.types.StructType

import org.apache.spark.sql.types.IntegerType

import org.apache.spark.sql.types.StringType

import org.apache.spark.sql.types.DoubleType

import org.apache.spark.sql.types.BooleanType

* **Creating a user defined scheme with StructType.**

val bank\_schema = new StructType()

.add("age",IntegerType,true)

.add("balance",IntegerType,true).add("campaign",IntegerType,true)

.add("contact",StringType,true)

.add("day",StringType,true)

.add("default",StringType,true)

.add("duration",StringType,true)

.add("education",StringType,true)

.add("housing",StringType,true)

.add("job",StringType,true)

.add("loan",StringType,true)

.add("marital",StringType,true)

.add("month",StringType,true)

.add("pdays",IntegerType,true)

.add("poutcome",StringType,true)

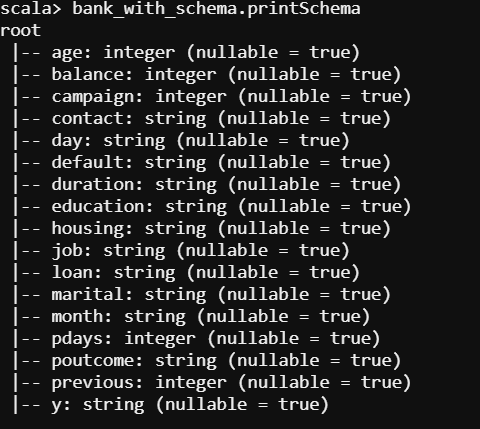
.add("previous",IntegerType,true)

.add("y",StringType,true)

* **Loading the data, schema, and options.**

val bank\_with\_schema = spark.read.format("json").option("multiline","true").schema(bank\_schema) .load("/user/prashanthteja049gmail/banking\_code.json")

* **Cross checking the schema.**

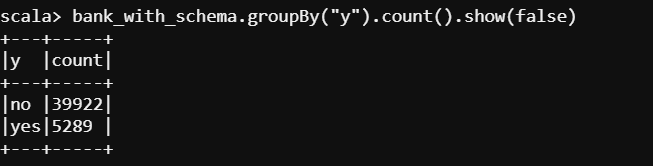


bank\_with\_schema.show(false)



* **Give marketing success rate and Failure Rate.**

bank\_with\_schema.groupBy("y").count().show(false)



* **Success Rate:**

val success\_rate = bank\_with\_schema.filter($"y" === "yes").count.toFloat / bank\_with\_schema.count.toFloat \*100

success\_rate: Float = 11.698481

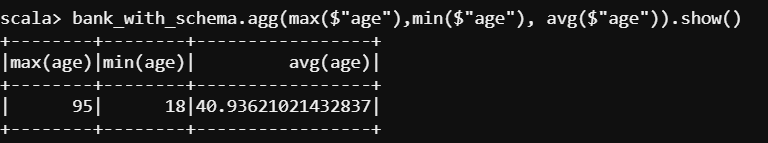
* **Failure Rate:**

val failure\_rate = bank\_with\_schema.filter($"y" === "no").count.toFloat / bank\_with\_schema.count.toFloat \*100

failure\_rate: Float = 88.30152

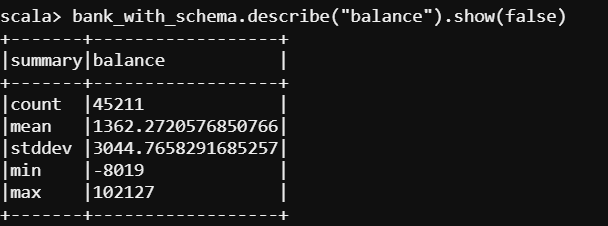
* **Give the maximum, mean, and minimum age of the average targeted customer**

bank\_with\_schema.agg(max($"age"),min($"age"), avg($"age")).show()



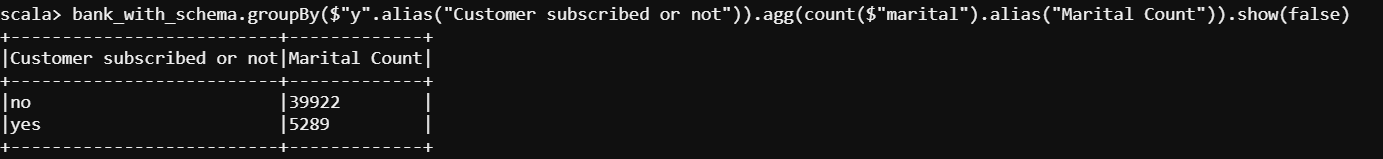
* **Check the quality of customers by checking average balance, median balance of customers**

bank\_with\_schema.describe("balance").show(false)



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

bank\_with\_schema.groupBy($"y".alias("Customer subscribed or not")).agg(count($"marital").alias("Marital Count")).show(false)



bank\_with\_schema.groupBy("marital","y").count.sort($"count").show()



**Here, we do Feature engineering using aggregate functions.**

val bank\_filter = bank\_with\_schema.filter($"y" === "yes")

* **Check if age matters in marketing subscription for deposit**

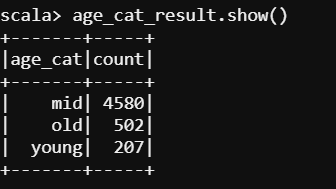
val age\_catageory = bank\_filter.withColumn("age\_cat", when($"age" < 25, "young")

.otherwise(when($"age" > 60, "old").otherwise("mid")))



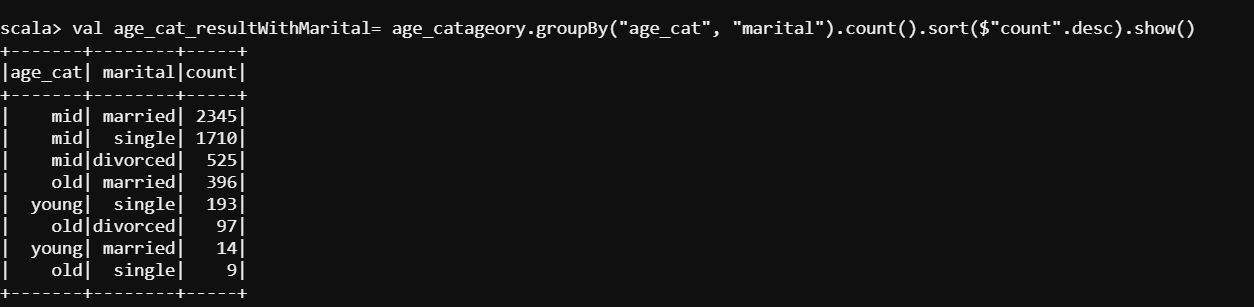
val age\_cat\_result = age\_catageory.groupBy("age\_cat").count()

age\_cat\_result.show()



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

val age\_cat\_resultWithMarital= age\_catageory.groupBy("age\_cat", "marital").count().sort($"count".desc).show()



**"mid" is the top sorted range with high subscribers.**

Hence,

So, we conclude that "mid" age people between the age **(25 years – 60 years)** are the targeted customers as the subscribers are high.