Source Code

```
val sqlContext = new org.apache.spark.sql.hive.HiveContext(sc)

val df =
sqlContext.read.format("com.databricks.spark.csv").option("header","true").option("inferSchema","true").option("delimiter",",").load("/user/karthikeyanm620_gmail/spark/Bank_Campaign_Data.csv")

df.printSchema();

df.show();


df.count();

df.filter($"y" === "yes").count()

df.filter($"y" === "no").count()


val success = df.filter($"y" === "yes")

success.agg(max($"age")).show()

success.agg(min($"age")).show()

success.agg(mean($"age")).show()


df.agg(mean($"balance")).show()

df.createOrReplaceTempView("data")

spark.sql("SELECT percentile_approx(balance, 0.5) FROM data").show()


df.groupBy("age").count().show();

spark.sql("select y, count(age) as younger_people from data where age<30 group by y").show();

spark.sql("select y, count(age) as people_30to40 from data where age between 30 and 40 group by y").show();
```

```
spark.sql("select y, count(age) as people_40to50 from data where age between 40 and 50 group by y").show();

spark.sql("select y, count(age) as people_50to60 from data where age between 50 and 60 group by y").show();

spark.sql("select y, count(age) as people_60to70 from data where age between 60 and 70 group by y").show();

spark.sql("select y, count(age) as oldest_people from data where age>80 group by y").show();


df.groupBy("marital").count().show()

spark.sql("select y, marital, count(age) as numbers from data group by y, marital").show();


success.createOrReplaceTempView("suc")

spark.sql("select marital, count(age) as younger_people from suc where age<30 group by marital").show();

spark.sql("select marital, count(age) as people_30to40 from suc where age between 30 and 40 group by marital").show();

spark.sql("select marital, count(age) as people_40to50 from suc where age between 40 and 50 group by marital").show();

spark.sql("select marital, count(age) as people_50to60 from suc where age between 50 and 60 group by marital").show();

spark.sql("select marital, count(age) as people_60to70 from suc where age between 60 and 70 group by marital").show();

spark.sql("select marital, count(age) as oldest_people from suc where age>80 group by marital").show();


spark.sql("select age, count(age) from suc group by age order by count(age) desc").show()
```

Solutions

1. Load data and create Spark data frame

The existing data in CSV format loaded in spark and created a data frame. The data has been viewed and schema also seen.

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

No. of people subscribed = 45211

Total no. of entries = 5289

Success rate = 0.1169

2a. Failure rate

No. of people not subscribed = 39922

Failure rate = 0.8830

3. Maximum, Mean, and Minimum age of average targeted customer.

Maximum age of targeted customer = 95

Mean age of targeted customer = 47.67

Minimum age of targeted customer = 18

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

Average balance of customers = 1362.27

Median balance of customers = 448

5. Check if age matters in marketing subscription for deposit

Yes. It was found that a particular aged group people subscribed more when compared others. People between the age of 30 and 45 had more subscription. Hence targeting these people will helps improve subscription rate.

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

Yes. From the analysis we found that married and single people contributes more for subscription. Especially married people subscribed more.

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

Yes. Both age and marital status together mattered for subscription. It was found that married people of age between 30 to 60 has more subscriptions. Also single people of age between 20 to 40 has more subscription.

8. Do feature engineering for column—age and find right age effect on campaign

The age column was analyzed and found out that the age group of 30 to 35 has more subscription rate. Especially, the age of 32 has more subscription.