

```

spark2-shell
--load csv
val mydf =
spark.read.option("delimiter", ";").option("header", "true").format("csv").load("/user/kasibanteg_gmail/banking.csv")

---show table columns and details
mydf.printSchema

----create temp view
mydf.createOrReplaceTempView("bankDomain")

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

val sucrate= mydf.filter('y=="yes").count.toDouble
val total= mydf.count.toDouble
val finalresult= sucrate/total *100

----failure rate
val failureRate= mydf.filter('y=="no").count.toDouble

val total= mydf.count.toDouble
val finalresult= failureRate/total *100

-----Give the maximum, mean, and minimum age of the average targeted customer
---using sql
val max_mean_avg = sql("select min(age), max(age), avg(age) from bankDomain").show()
--using scala
val max_mean_avg = mydf.select(max($"age"), min($"age"), avg($"age")).show()

----Check the quality of customers by checking average balance, median balance of customer
--average
mydf.select(avg($"balance")).show()

---median
val median = sql("SELECT percentile_approx(balance, 0.5) FROM bankDomain").show()

-----Check if age matters in marketing subscription for deposit
val age = sql("select age, count(*) as number from bankDomain where y='yes' group by age order by number desc ").show()

----Check if marital status mattered for a subscription to deposit
val marital = sql("select marital, count(*) as number from bankDomain where y='yes' group by marital order by number desc ").show()

```

```
-----Check if age and marital status together mattered for a subscription to  
deposit scheme  
val age_marital = sql("select age, marital, count(*) as number from bankDomain where  
y='yes' group by age, marital order by number desc ").show()
```

```
---Do feature engineering for the bank and find the right age effect on the  
campaign  
val feature_column=sql("select 'Young', count(*) from bankDomain where y = 'yes'  
and age < 25 union select 'Middle Aged', count(*) from bankDomain where y = 'yes'  
and age >= 25 and age <=60 union select 'Old', count(*) from bankDomain where y=  
'yes' and age >60 ").show()
```