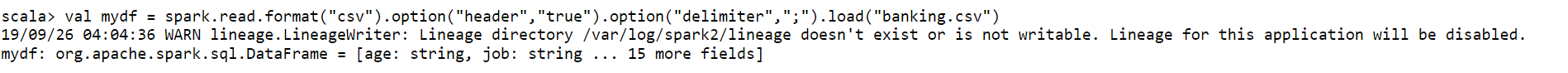
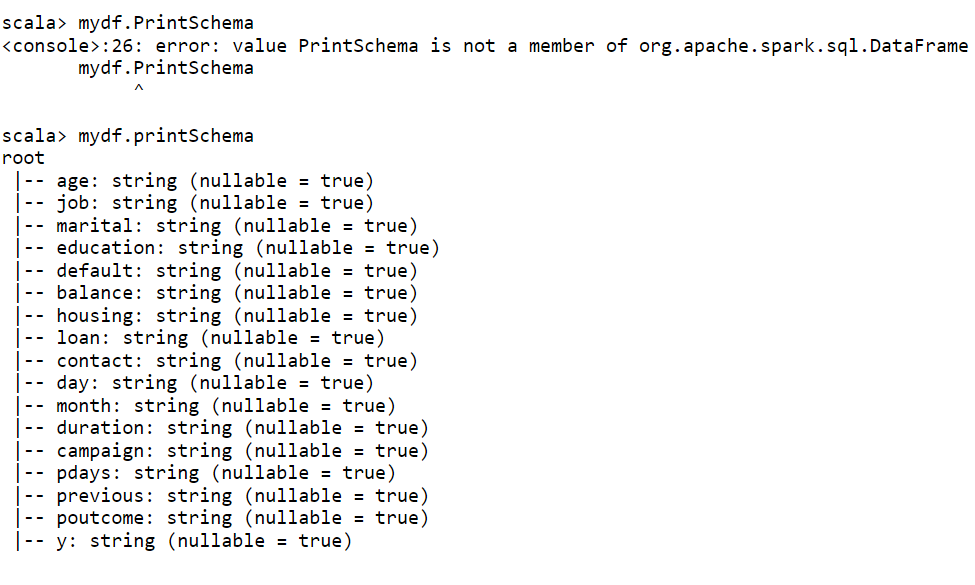
4. Market Analysis in Banking Domain

Step1:

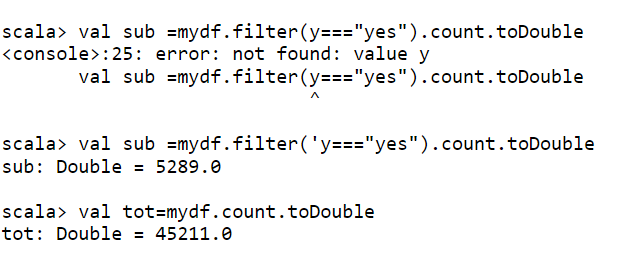
val mydf = spark.read.format("csv").option("header","true").option("delimiter",";").load("banking.csv")

result:



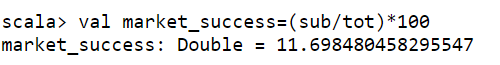


step2:

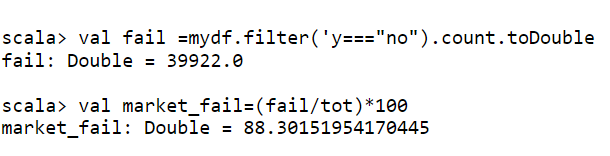


val market\_success=(sub/tot)\*100

output:

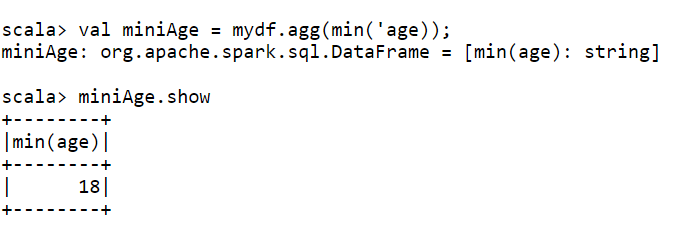


val fail =mydf.filter('y==="no").count.toDouble



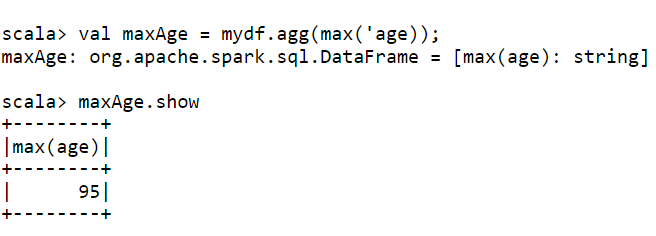
3. minimum age:

val miniAge = mydf.agg(min('age));



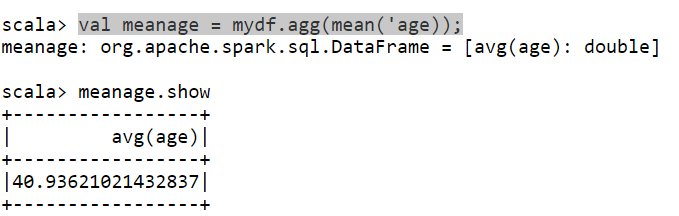
Maximum age:

val maxAge = mydf.agg(max('age));



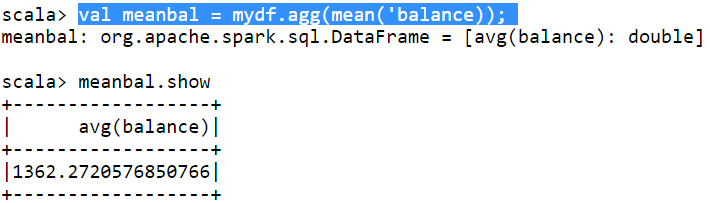
Mean age:

val meanage = mydf.agg(mean('age));

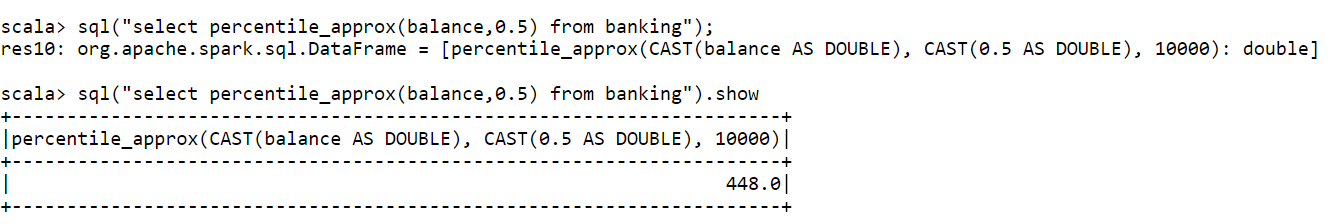


4.mean balance:

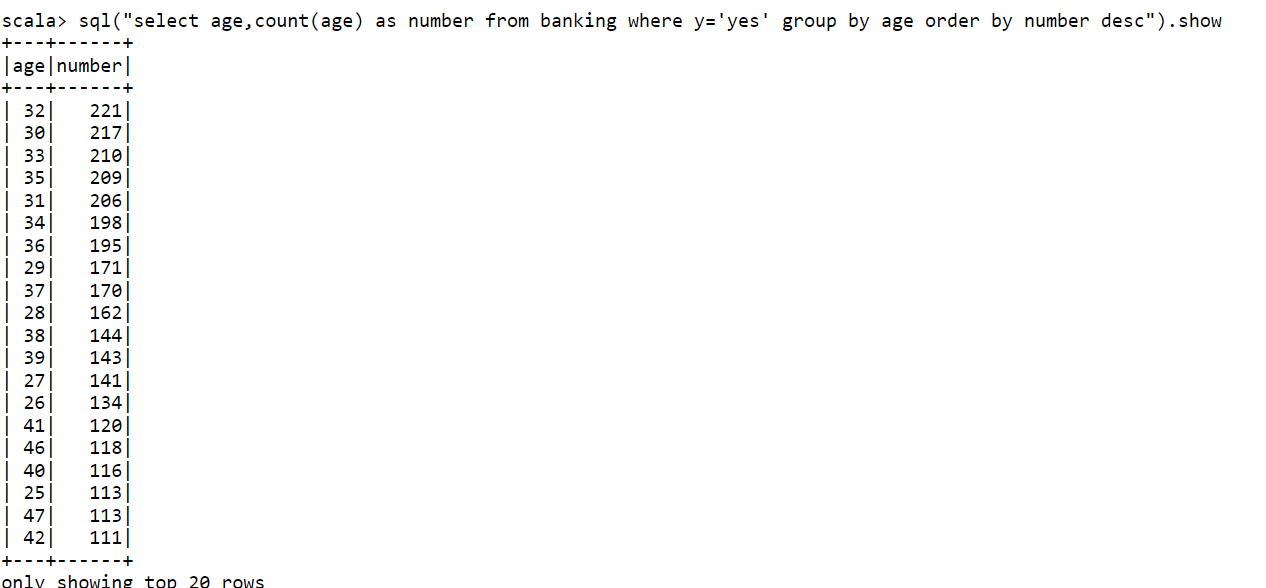
val meanbal = mydf.agg(mean('balance));

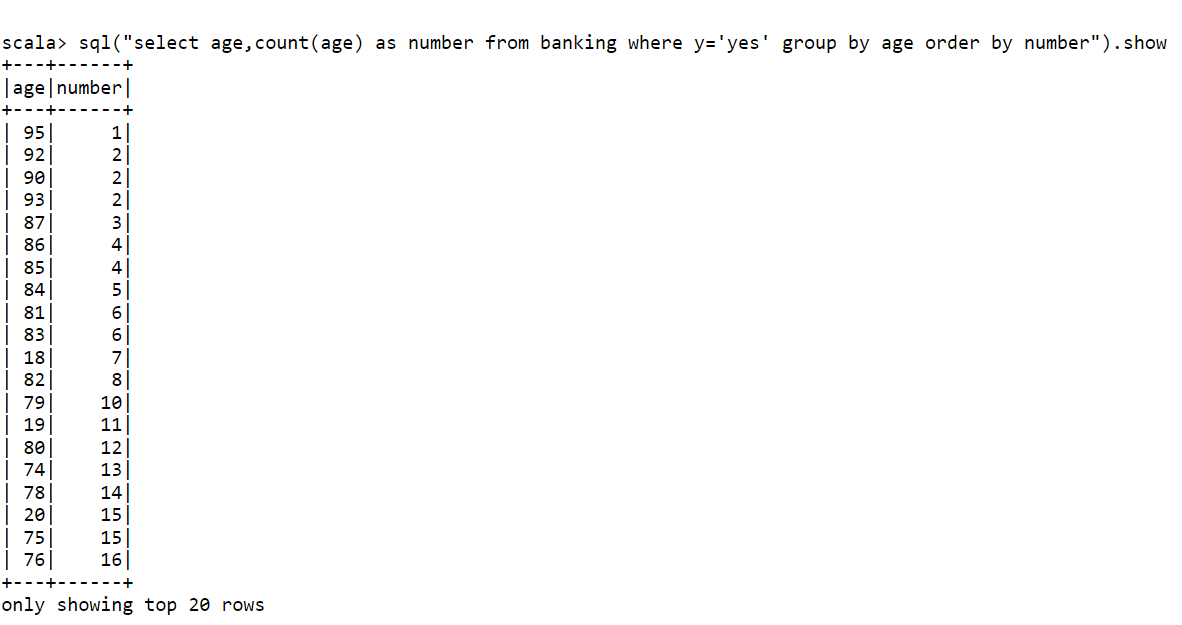


Median balance:



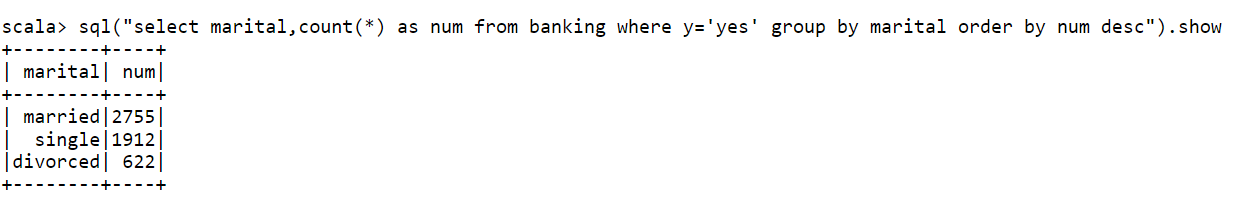
4.checking if age matters:





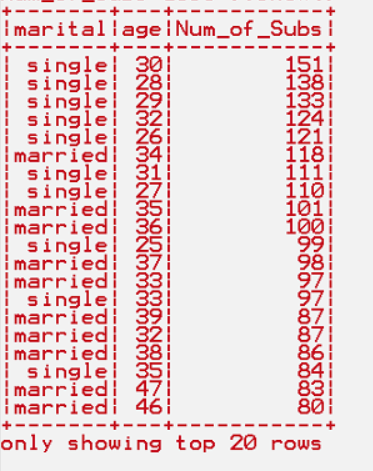
Yes age matters.

5.checking if marriage matters:



Yes it affects.

6.checking age, marriage both matters:

sql("select age,marital,count(\*) as num from banking where y='yes' group by marital,age order by num desc ").show 

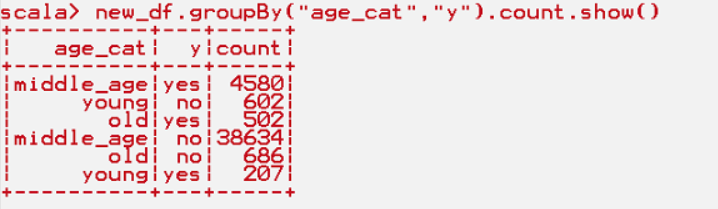
this tells us that both are important to get more subscriptions

7. doing feature enginerring :

val new\_df=mydf.withColumn("age\_cat",when('age<25,"young").otherwise(when('age>60,"old").otherwise("middle\_age")))

grouping by age\_cat:

new\_df.groupBy("age\_cat","y").count.show



The out put is showing that the middle aged persons are subscribed more than any other (below 25,above 60).