|  |  |
| --- | --- |
| **ANALYISIS OF MARKETING DATA FOR CALL CAMPAIGN BY BANK** | **Brief description of project work :**  Banking institution—Ran a marketing campaign to convince potential customers to invest in bank term deposit. |

**Problem Statement’s:**

**QUESTION  1. Load data and create Spark data frame**

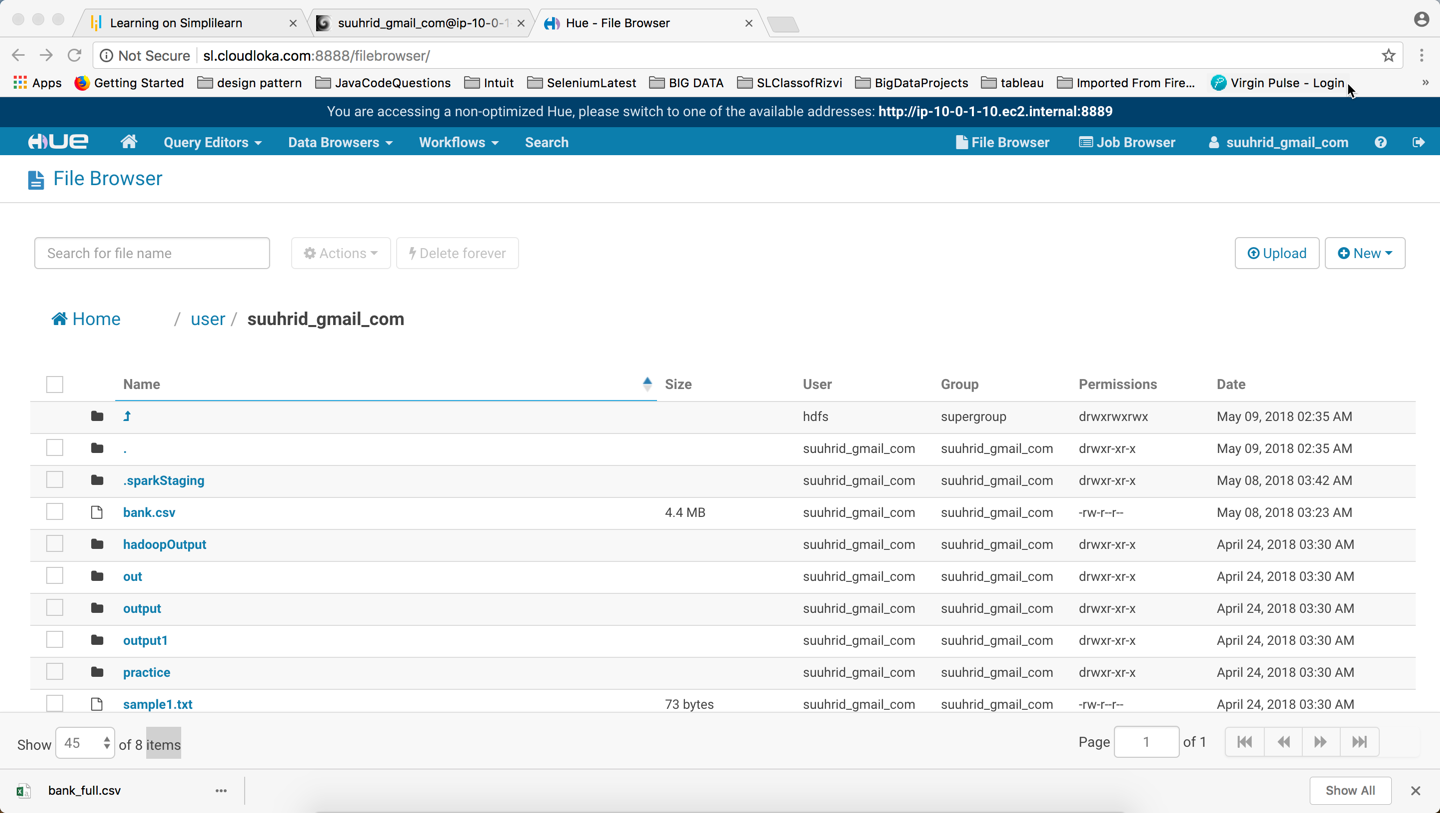
*val bankDataSet = sc.textFile("/user/suuhrid\_gmail\_com/bank.csv");*

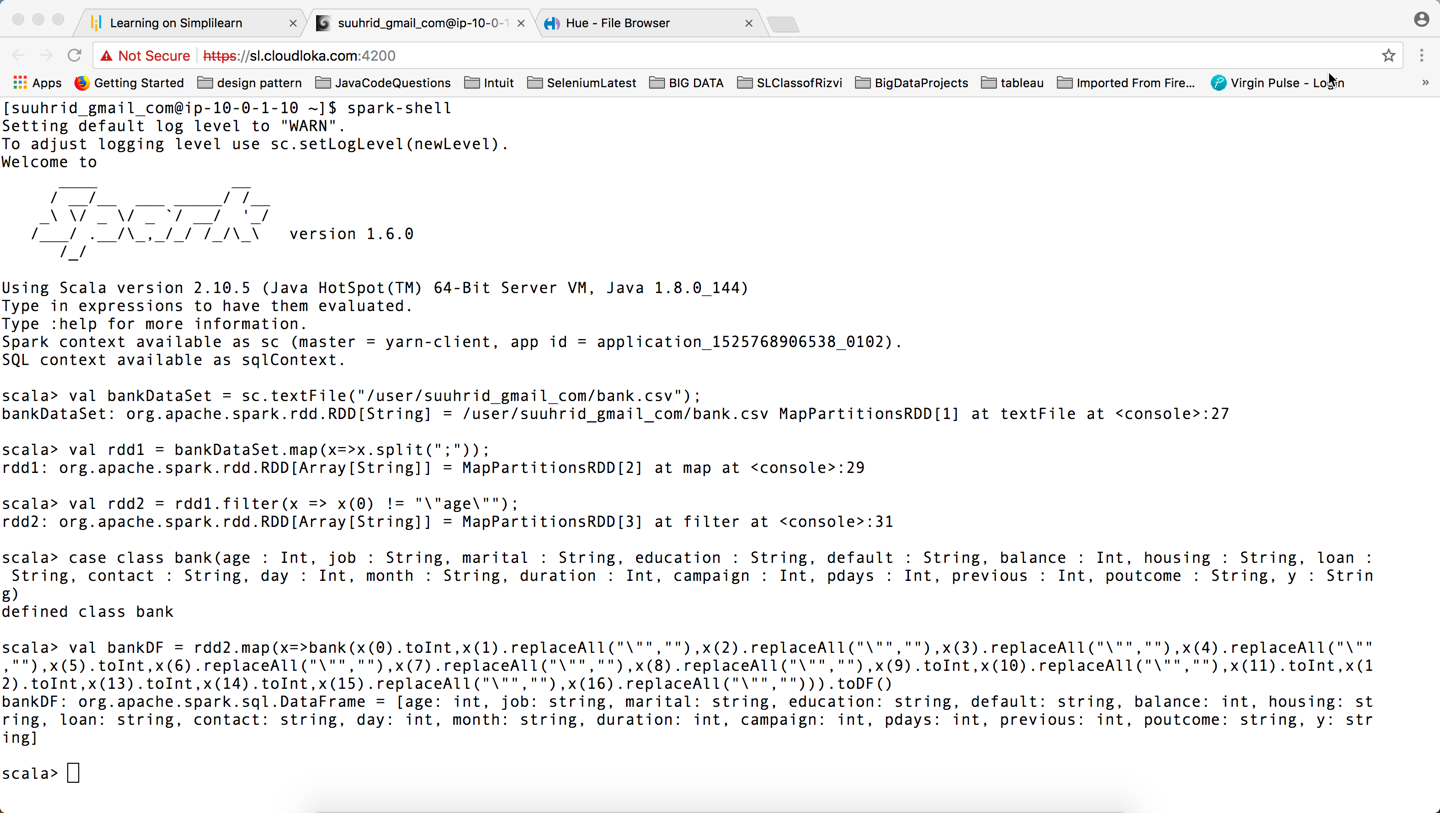
*val rdd1 = bankDataSet.map(x=>x.split(";"));*

*val rdd2 = rdd1.filter(x => x(0) != "\"age\"");*

*case class bank(age : Int, job : String, marital : String, education : String, default : String, balance : Int, housing : String, loan : String, contact : String, day : Int, month : String, duration : Int, campaign : Int, pdays : Int, previous : Int, poutcome : String, y : String)*

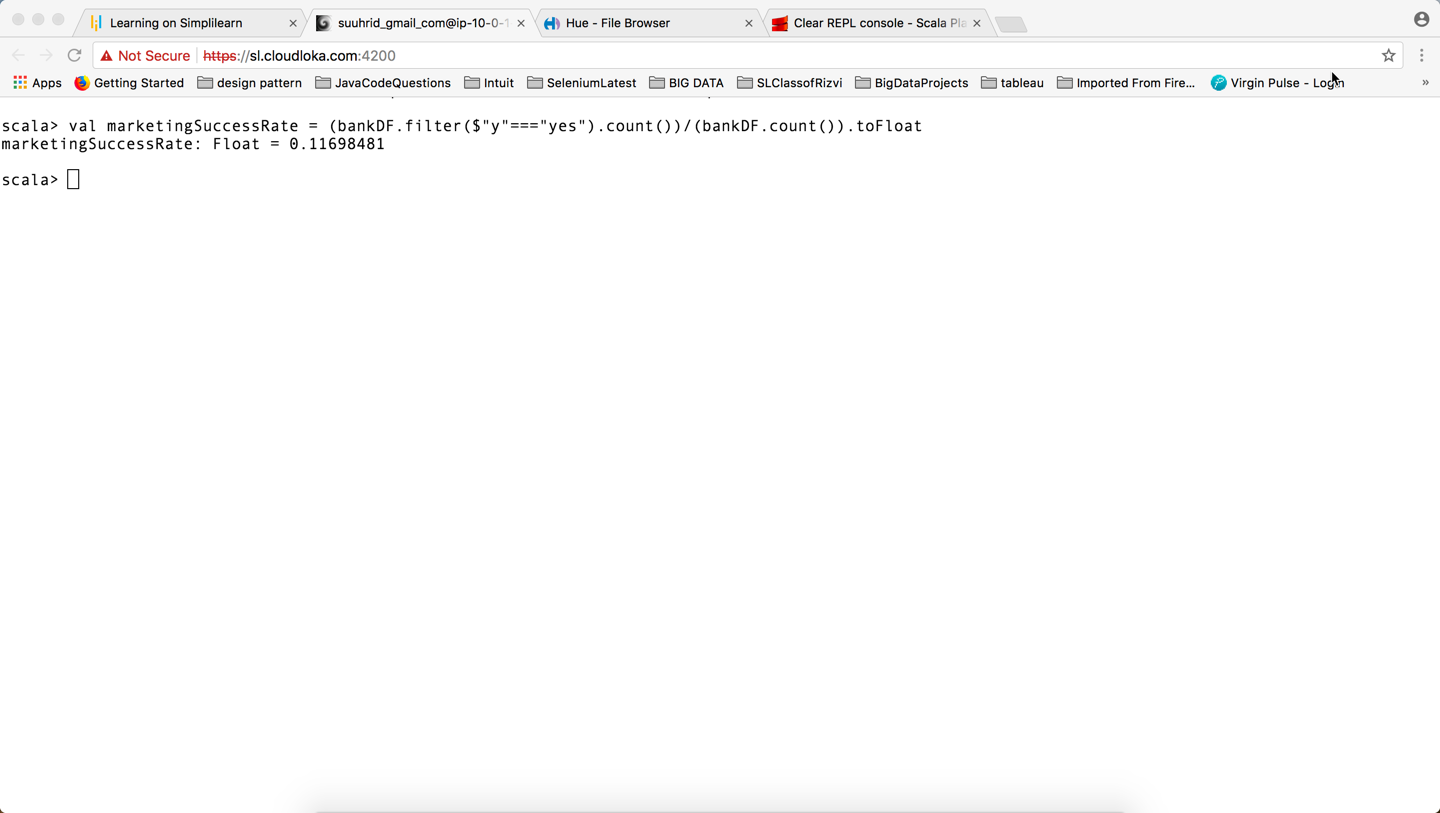
*val bankDF = rdd2.map(x=>bank(x(0).toInt,x(1).replaceAll("\"",""),x(2).replaceAll("\"",""),x(3).replaceAll("\"",""),x(4).replaceAll("\"",""),x(5).toInt,x(6).replaceAll("\"",""),x(7).replaceAll("\"",""),x(8).replaceAll("\"",""),x(9).toInt,x(10).replaceAll("\"",""),x(11).toInt,x(12).toInt,x(13).toInt,x(14).toInt,x(15).replaceAll("\"",""),x(16).replaceAll("\"",""))).toDF()*



**

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

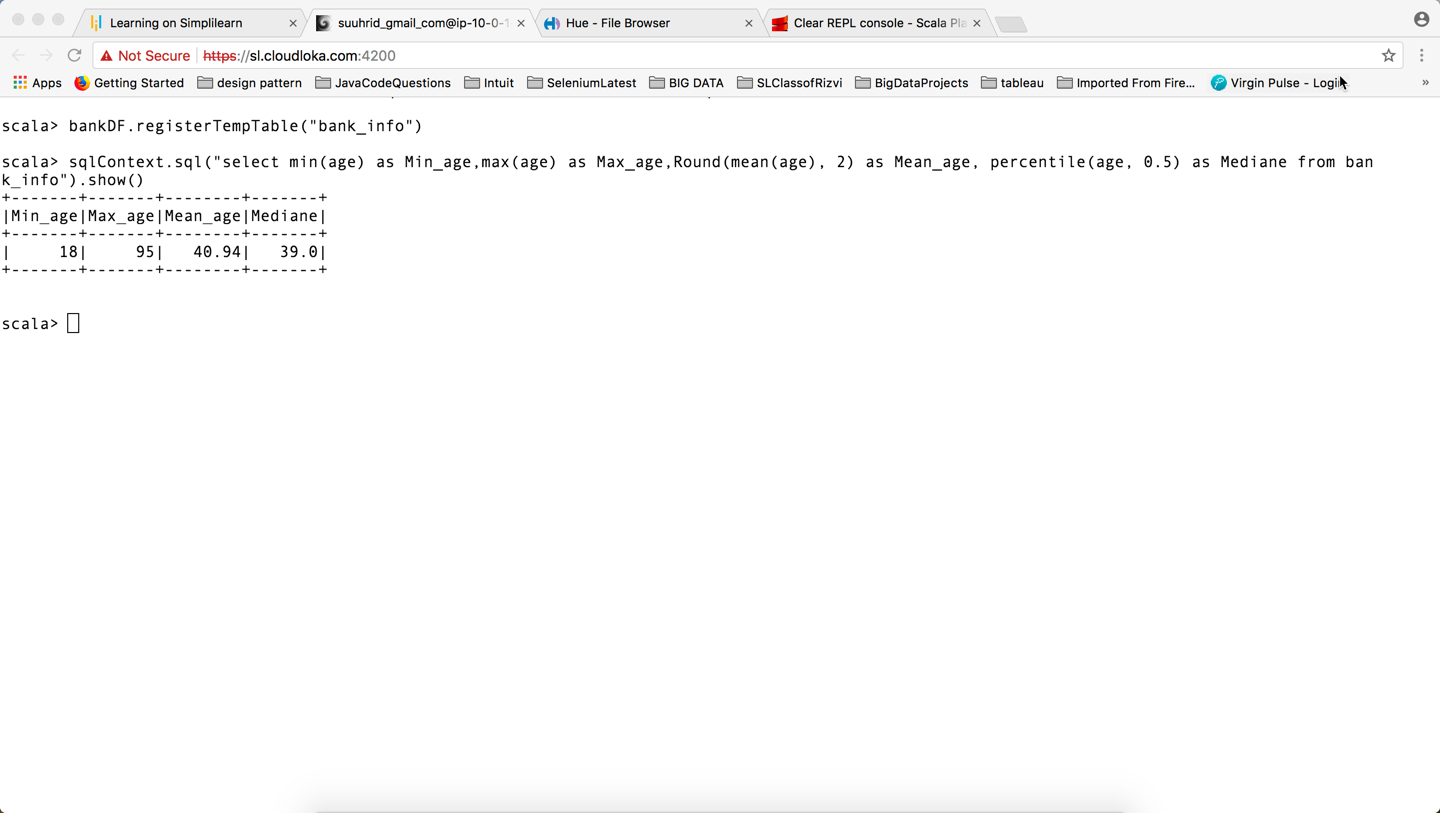
*val marketingSuccessRate = (bankDF.filter($"y"==="yes").count())/(bankDF.count()).toFloat*



**QUESTION  3 - Check max, min, Mean and median age of average targeted customer.**

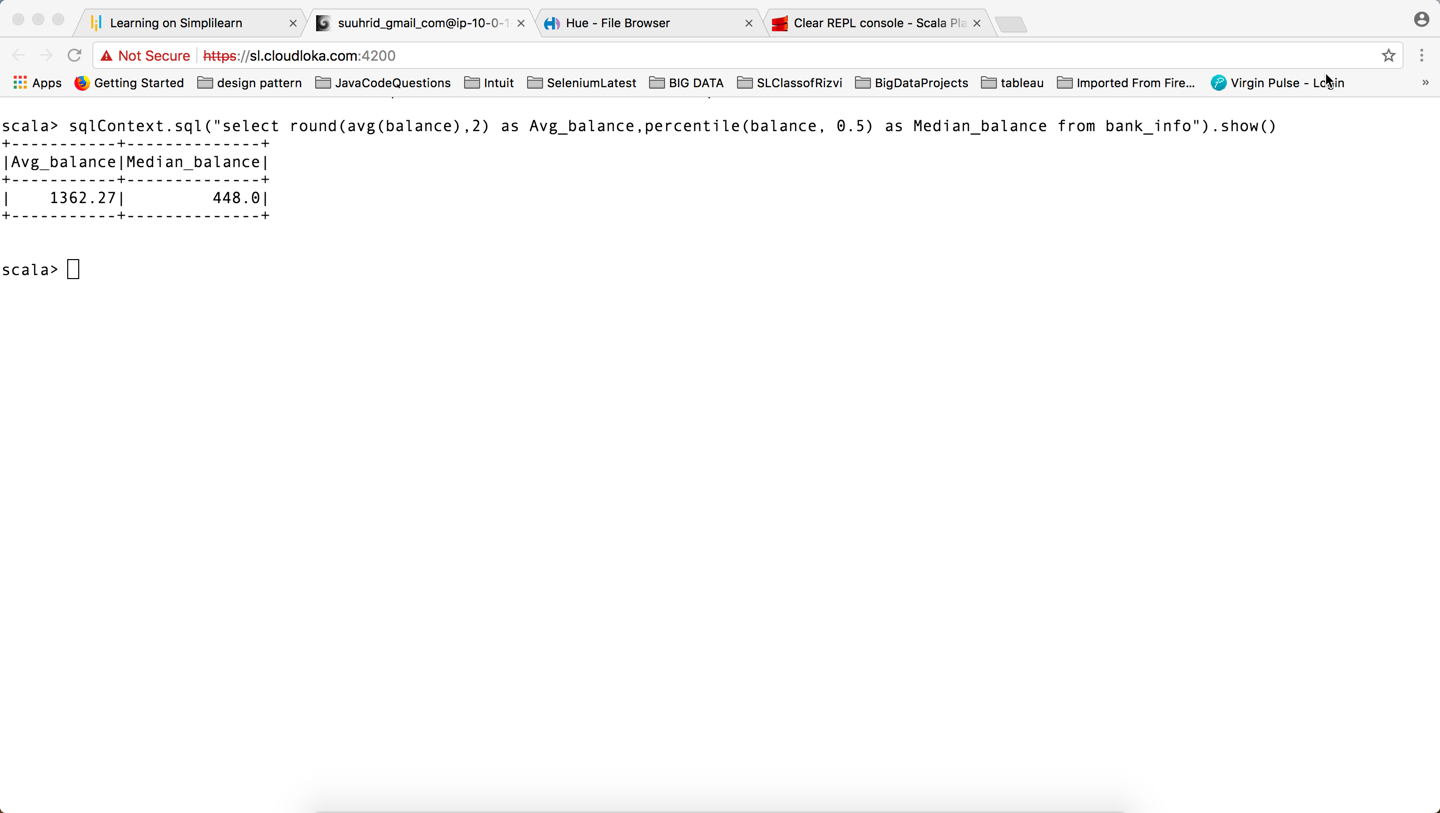
bankDF.registerTempTable("bank\_info")

sqlContext.sql("select min(age) as Min\_age,max(age) as Max\_age,Round(mean(age), 2) as Mean\_age, percentile(age, 0.5) as Mediane from bank\_info").show()

****

**QUESTION  4 - Check quality of clients by checking average balance, median balance of clients.**

sqlContext.sql("select round(avg(balance),2) as Avg\_balance,percentile(balance, 0.5) as Median\_balance from bank\_info").show()

****

**QUESTION  5 – Check if age matters in marketing subscription for deposit.**

sqlContext.sql("select avg(age) as avg\_age,y as subscription from bank\_info group by y").show()

****

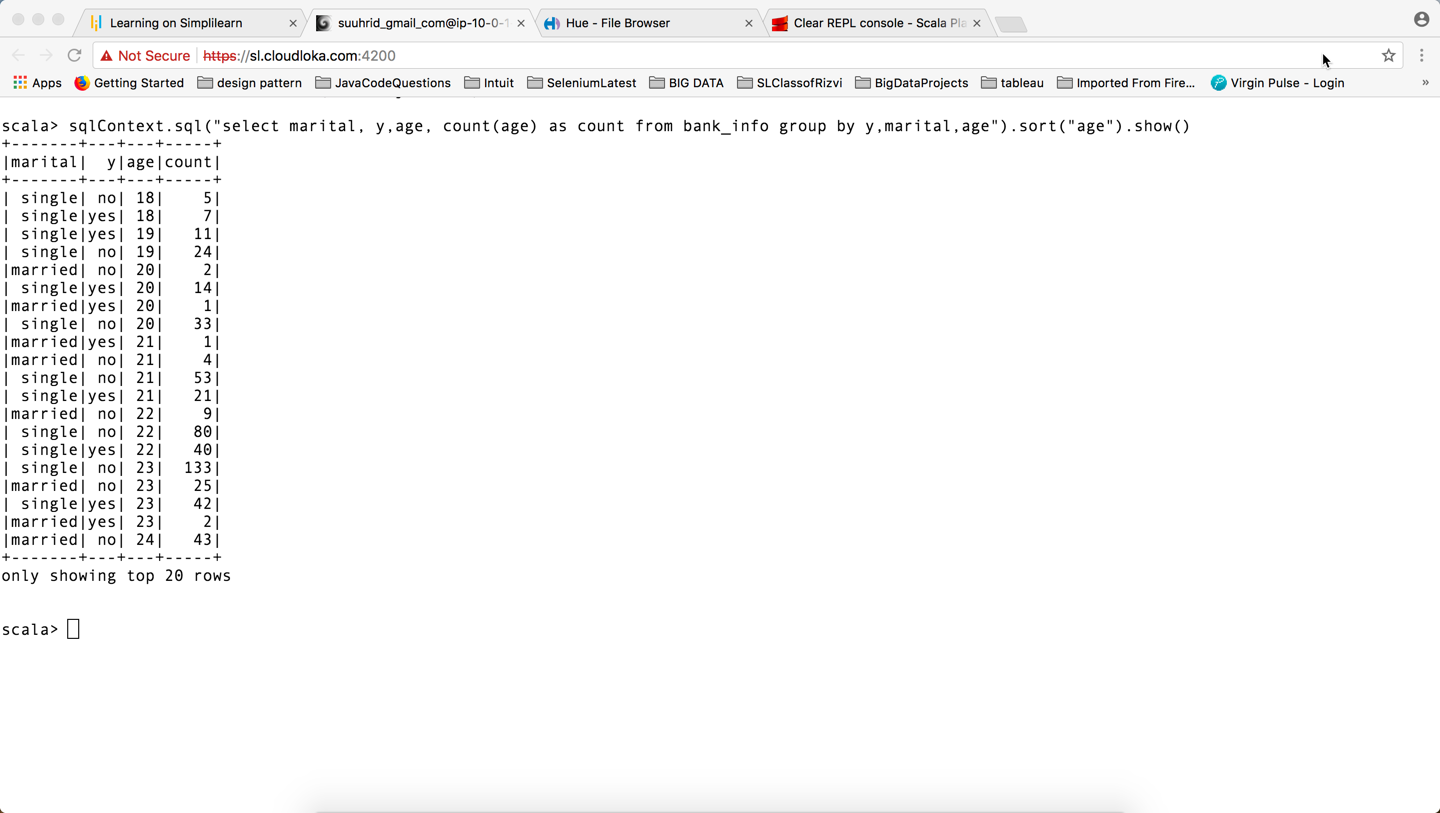
**QUESTION  6 – Check if marital status mattered for subscription to deposit**

sqlContext.sql("select marital, y,count(\*) as count from bank\_info group by y,marital").show()

****

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

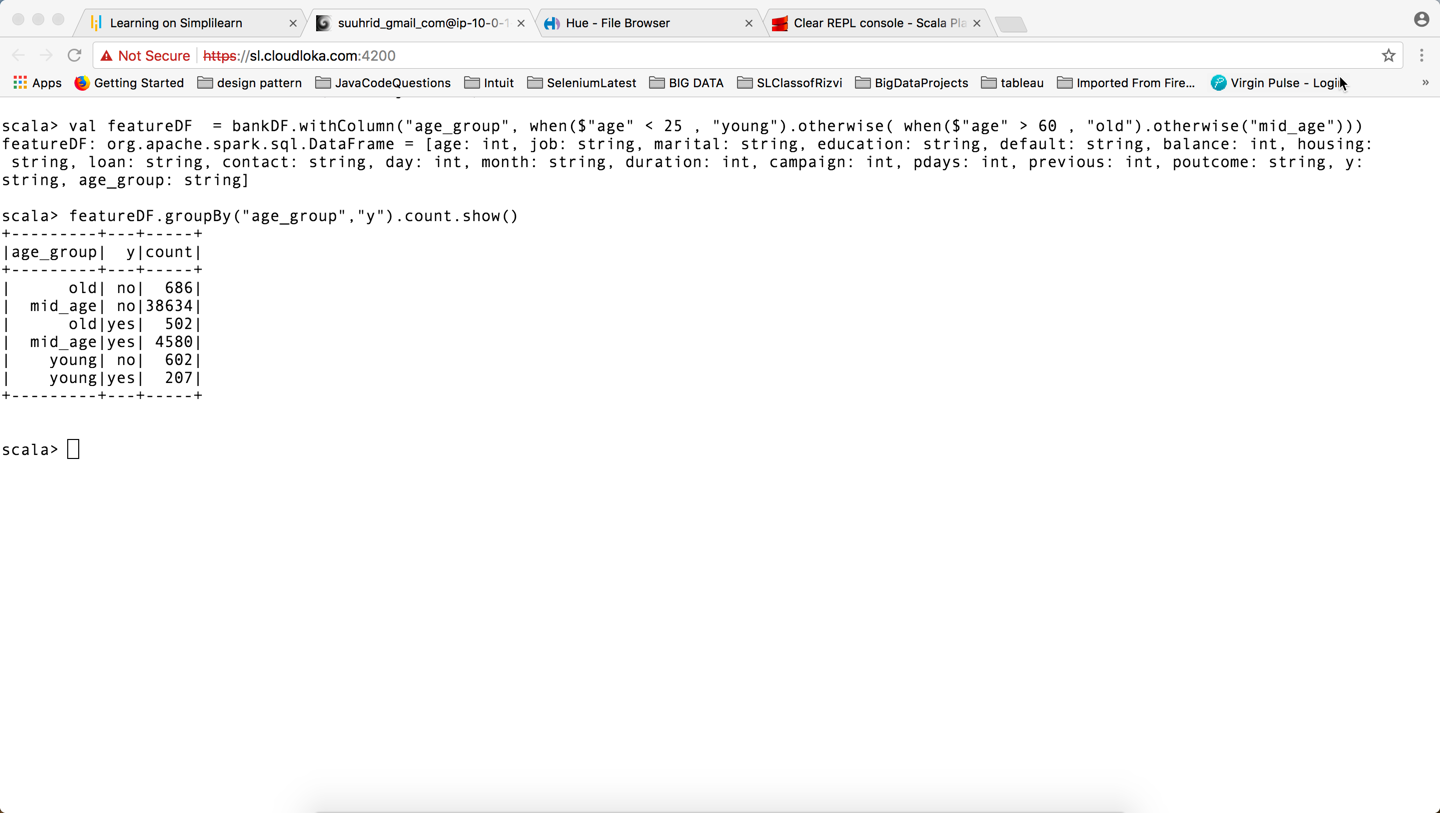
sqlContext.sql("select marital, y,age, count(age) as count from bank\_info group by y,marital,age").sort("age").show()

****

**QUESTION  8 – Do Feature engineering for age column and find right age effect on campaign.**

val featureDF  = bankDF.withColumn("age\_group", when($"age" < 25 , "young").otherwise( when($"age" > 60 , "old").otherwise("mid\_age")))

featureDF.groupBy("age\_group","y").count.show()

**

**Note: Most of the people who are between the age 25-60 opted for the subscription.**