**Market Analysis in Banking Domain**

**Mothe shivaRaj**

**It is used to send the file from Local to User**

Hdfs dfs -copyFromLocal /home/mothe\_shivaraj09\_gmail/market.txt /user/mothe\_shivaraj09\_gmail

**To Enable the Spark-shell**

1) spark-shell --packages com.databricks: spark-csv\_2.11:1.5.0

**Load data and create a Spark data frame**

2) Val df = sqlContext.read.format ("com.databricks.spark.csv").option ("header","true").option("inferSchema","true").

option("delimiter",",").load("bank\_new.csv")

3) val totalcount = df.count().toDouble

45211.0

4) val subscription\_count= df.filter($"y" === "yes").count().toDouble

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

5) val success\_rate = subscription\_count/totalcount

**Give Marketing Failure rate**

val subscription\_count1= df.filter($"y" === "no").count().toDouble

val failure\_rate = subscription\_count1/totalcount

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

6) df.select(max($"age"), avg($"age"), min($"age")).show

7) df.registerTempTable("bankdetails")

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

8) sqlContext.sql("select percentile(balance,0.5) as median ,avg(balance) as average from bankdetails").show

9) df.groupBy("y").agg(avg($"age")).show

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

10) df.groupBy("y").agg(count($"marital")).show

Check if marital status mattered for a subscription to deposit

11) df.groupBy("marital","y").count().sort($"count".desc).show

12) df.groupBy("age","y").count().sort($"count".desc).show

13) df.groupBy("age","y").count().sort($"count".desc).count

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

14) import org.apache.spark.sql.functions.udf

def ageToCategory = udf((age:Int) => { age match {

case t if t < 30 => "young" case t if t > 65 => "old" case \_ => "mid" } } )

**Do feature engineering for the bank and find the right age effect on the campaign.**

15) val newdf = df.withColumn(“agecategory”,ageToCategory(df(“age”)))

16) newdf.groupBy("agecategory","y").count().sort($"count".desc).show