**SOURCE CODE**

val bank\_people\_data = spark.read.option("multiline","true").json("/user/sowmyas1096\_gmail/bank\_edited.json");

bank\_people\_data.show()

bank\_people\_data.registerTempTable("datanewtable")

bank\_people\_data.select(max($"age")).show()

bank\_people\_data.select(min($"age")).show()

bank\_people\_data.select(avg($"age")).show()

bank\_people\_data.select(avg($"balance")).show()

val median = spark.sql("SELECT percentile\_approx(balance, 0.5) FROM datanewtable").show()

val agedata = spark.sql("select age, count(\*) as number from datanewtable where y='yes' group by age order by number desc")

agedata.show()

val maritaldata = spark.sql("select marital, count(\*) as number from datanewtable where y='yes' group by marital order by number desc")

maritaldata.show()

val ageandmaritaldata = spark.sql("select age, marital, count(\*) as number from datanewtable where y='yes' group by age,marital order by number desc")

ageandmaritaldata.show()

val agedata = spark.udf.register("agedata",(age:Int) => {

if (age < 20)

"Teen"

else if (age > 20 && age <= 32)

"Young"

else if (age > 33 && age <= 55)

"Middle Aged"

else

"old"

})

//Replacing the old age column with the new age column

val banknewDF = bank\_people\_data.withColumn("age",agedata(bank\_people\_data("age")))

banknewDF.show()

banknewDF.registerTempTable("banknewtable")

//which age group subscribed the most

val targetage = spark.sql("select age, count(\*) as number from banknewtable where y='yes' group by age order by number desc")

targetage.show()

//pipelining with string Indexer

import org.apache.spark.ml.feature.StringIndexer

val agedata2 = new StringIndexer().setInputCol("age").setOutputCol("ageindex")

//Fitting the model

var strindModel = agedata2.fit(banknewDF)

//assigns generated value of index of the column, by feature engineering

strindModel.transform(banknewDF).select("age","ageIndex").show(5)