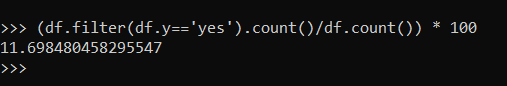
1. Create DF and load data

file = "/home/miles/futurense\_hadoop-pyspark/labs/dataset/bankmarket/bankmarketdata.csv"

df=spark.read.options(header=True,delimiter=';',inferSchema=True).csv(file)

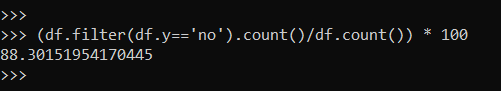
2.Give marketing success rate. (No. of people subscribed / total no. of entries)

(df.filter(df.y=='yes').count()/df.count()) \* 100



3.Give marketing failure rate

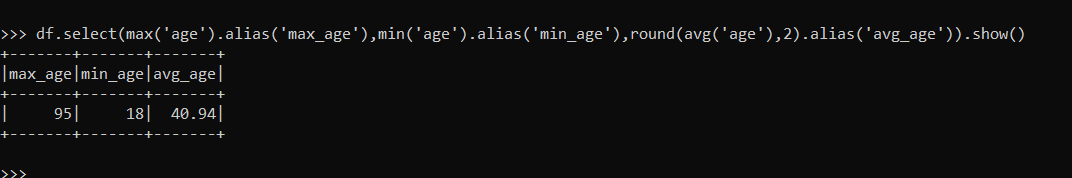
(df.filter(df.y=='no').count()/df.count()) \* 100



4.Maximum, Mean, and Minimum age of the average targeted customer

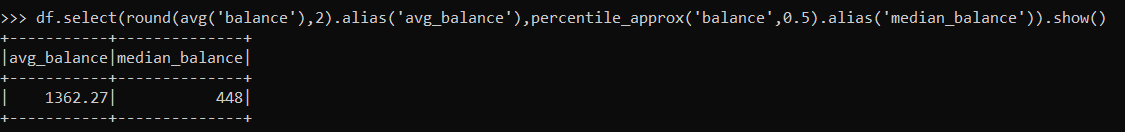
from pyspark.sql.functions import \*

df.select(max('age').alias('max\_age'),min('age').alias('min\_age'),round(avg('age'),2).alias('avg\_age')).show()



5.Check the quality of customers by checking the average balance, median balance of customers

df.select(round(avg('balance'),2).alias('avg\_balance'),percentile\_approx('balance',0.5).alias('median\_balance')).show()



6. Check if age matters in marketing subscription for deposit

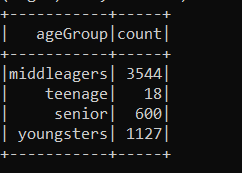
df.filter(df.y=='yes').select('age').groupBy('age').count().orderBy('count',ascending=False).show()



7. Show AgeGroup [Teenagers, Youngsters, MiddleAgers, Seniors] wise Subscription Count.

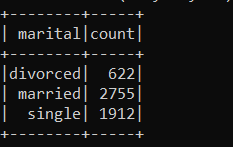
df.filter(df.y=='yes').select('age',when((col('age')>=13) & (col('age')<=19),'teenage').when((col('age')>=20) & (col('age')<=30),'youngsters').when((col('age')>=31) & (col('age')<=59),'middleagers').when(col

('age')>=60,'senior').alias('ageGroup')).groupBy('ageGroup').count().show()



8. Check if marital status mattered for subscription to deposit.

df.filter(df.y=='yes').groupBy('marital').count().show()



9. Check if age and marital status together mattered for subscription to deposit scheme

df.filter(df.y=='yes').groupBy('age','marital').count().orderBy('count',ascending=False).show()

