**Bank Marketing Campaign Data Analysis with PySpark SQL**

from pyspark.sql import SparkSession

spark = SparkSession.builder.master("local[1]") \

.appName('SparkByExamples.com').getOrCreate()

from pyspark.sql.functions import \*

Bank\_DF = spark.read.format("csv").load("/mnt/c/Users/miles.MILE-BL-4409-LA/futurense\_hadoop-pyspark/labs/dataset/bankmarket/bankmarketdata.csv", header=True, sep = ";", escape = ",", inferSchema = True )

Bank\_DF.registerTempTable("Bank")

output=spark.sql("select (case when age<13 then 'Kids' when age<20 then 'Teenagers' \

when age < 31 then 'Young' \

when age<50 then 'MiddleAgers' else 'Seniors' end) as peopletype,count(age) from bank where y='yes' group by peopletype order by count(age)")

output.write.parquet("hdfs://localhost:9000/user/training/bankmarketing/out/parquet12")

data=spark.read.parquet("hdfs://localhost:9000/user/training/bankmarketing/out/parquet12")

data.show()

output1=spark.sql("select (case when age<13 then 'Kids' when age<20 then 'Teenagers' \

when age < 31 then 'Young' \

when age<50 then 'MiddleAgers' else 'Seniors' end) as peopletype,count(age) as age from bank where y='yes' group by peopletype having count(age)>2000")

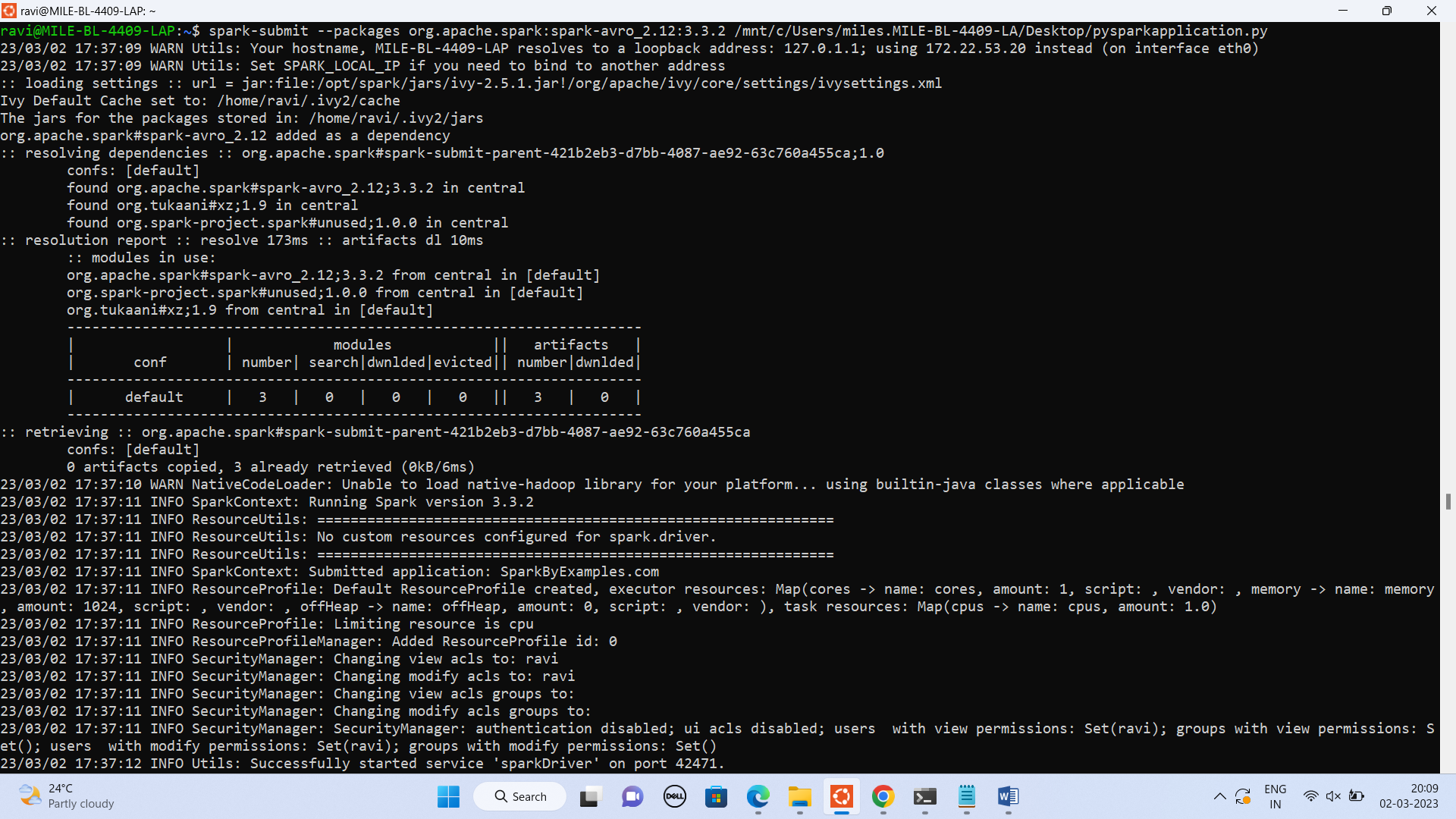
output1.select("peopletype","age").write.format("avro").save("hdfs://localhost:9000/user/training/bankmarketing/out/avro2")

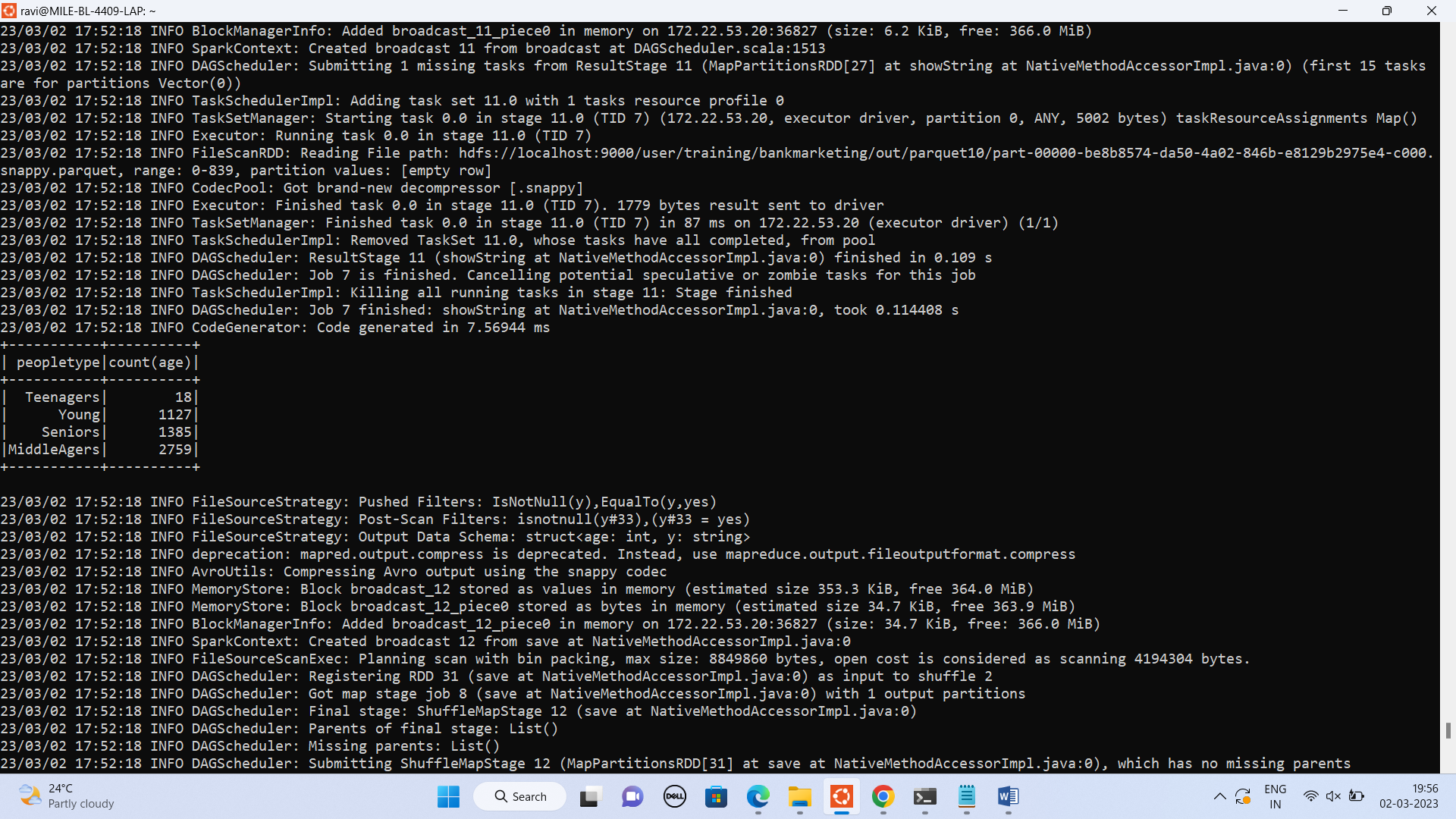
data1=spark.read.format("avro").load("hdfs://localhost:9000/user/training/bankmarketing/out/avro2")

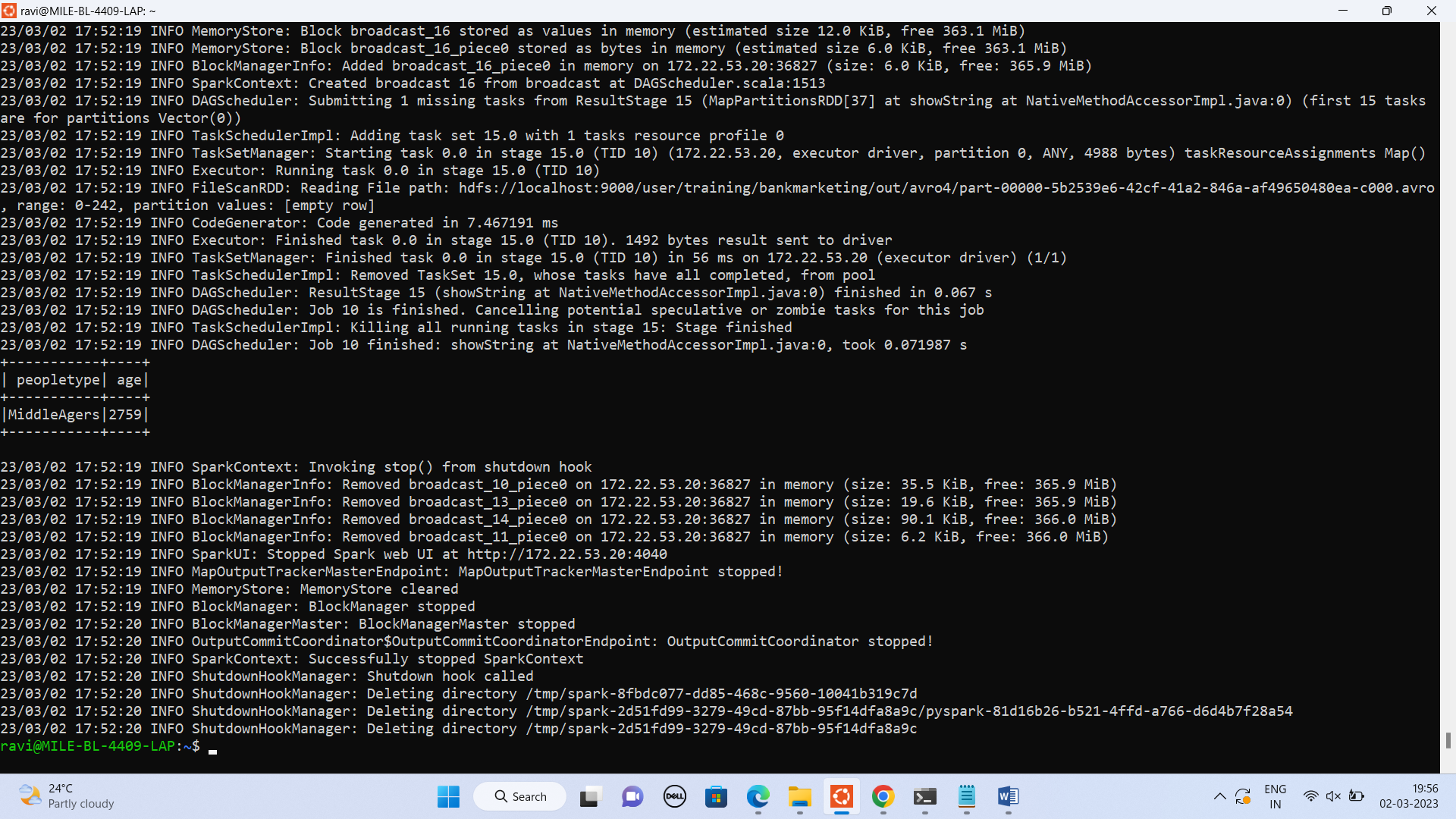
data1.show()

**# 1] spark-submit bank-marketing-analysis.py => Runs in local mode**

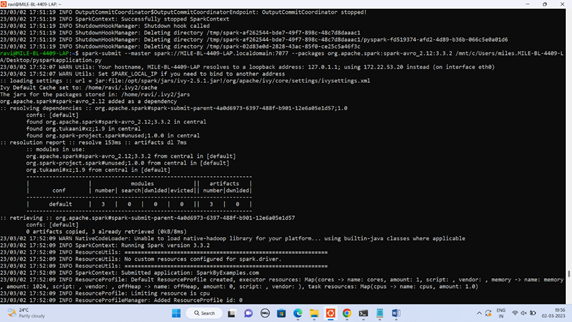
**spark-submit --packages org.apache.spark:spark-avro\_2.12:3.3.2 /mnt/c/Users/miles.MILE-BL-4409-LA/Desktop/pysparkapplication.py**







**2] Run in cluster mode:**

**spark-submit --master spark://MILE-BL-4409-LAP.localdomain:7077 --packages org.apache.spark:spark-avro\_2.12:3.3.2 /mnt/c/Users/miles.MILE-BL-4409-LA/Desktop/pysparkapplication.p**

