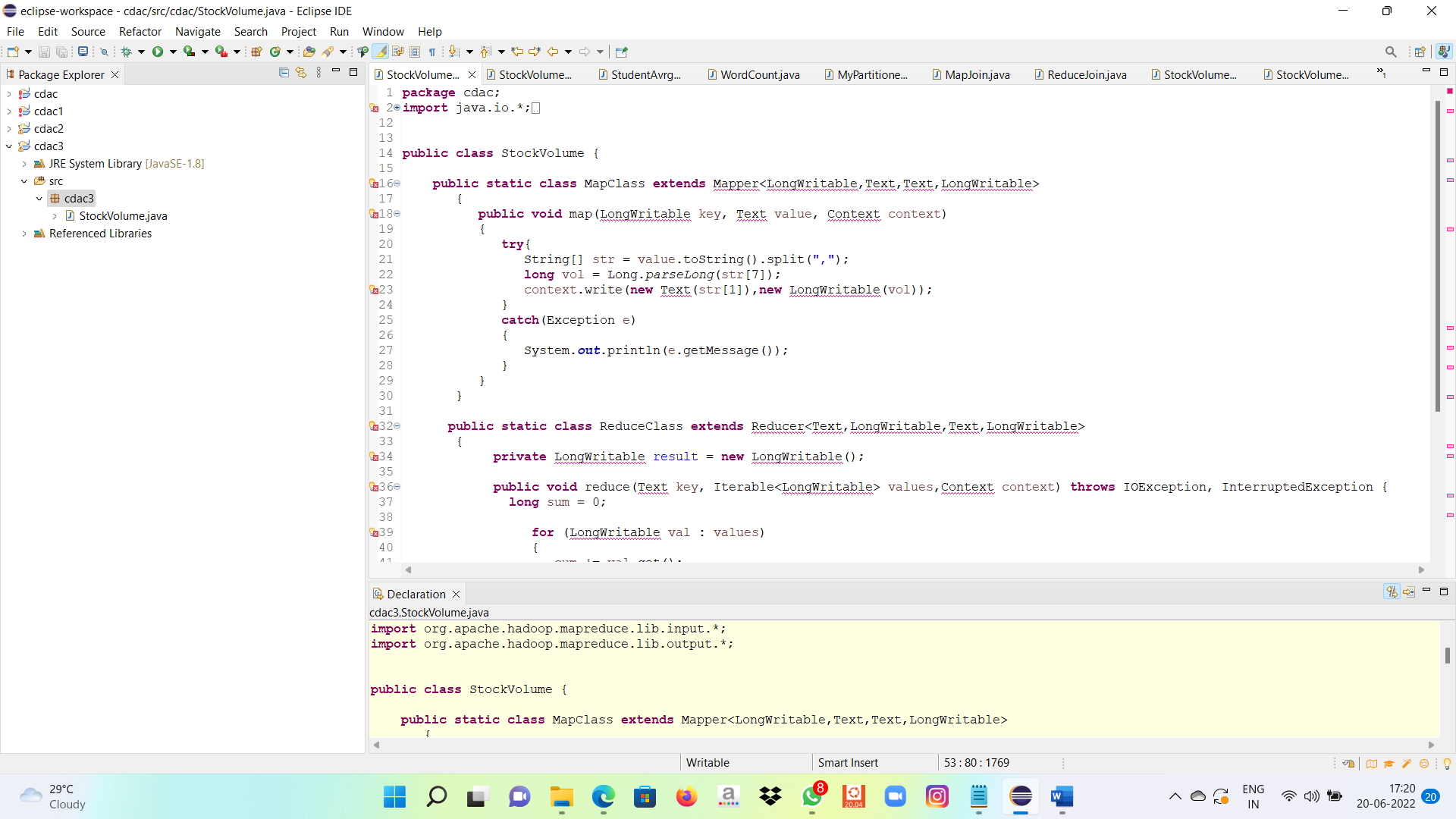
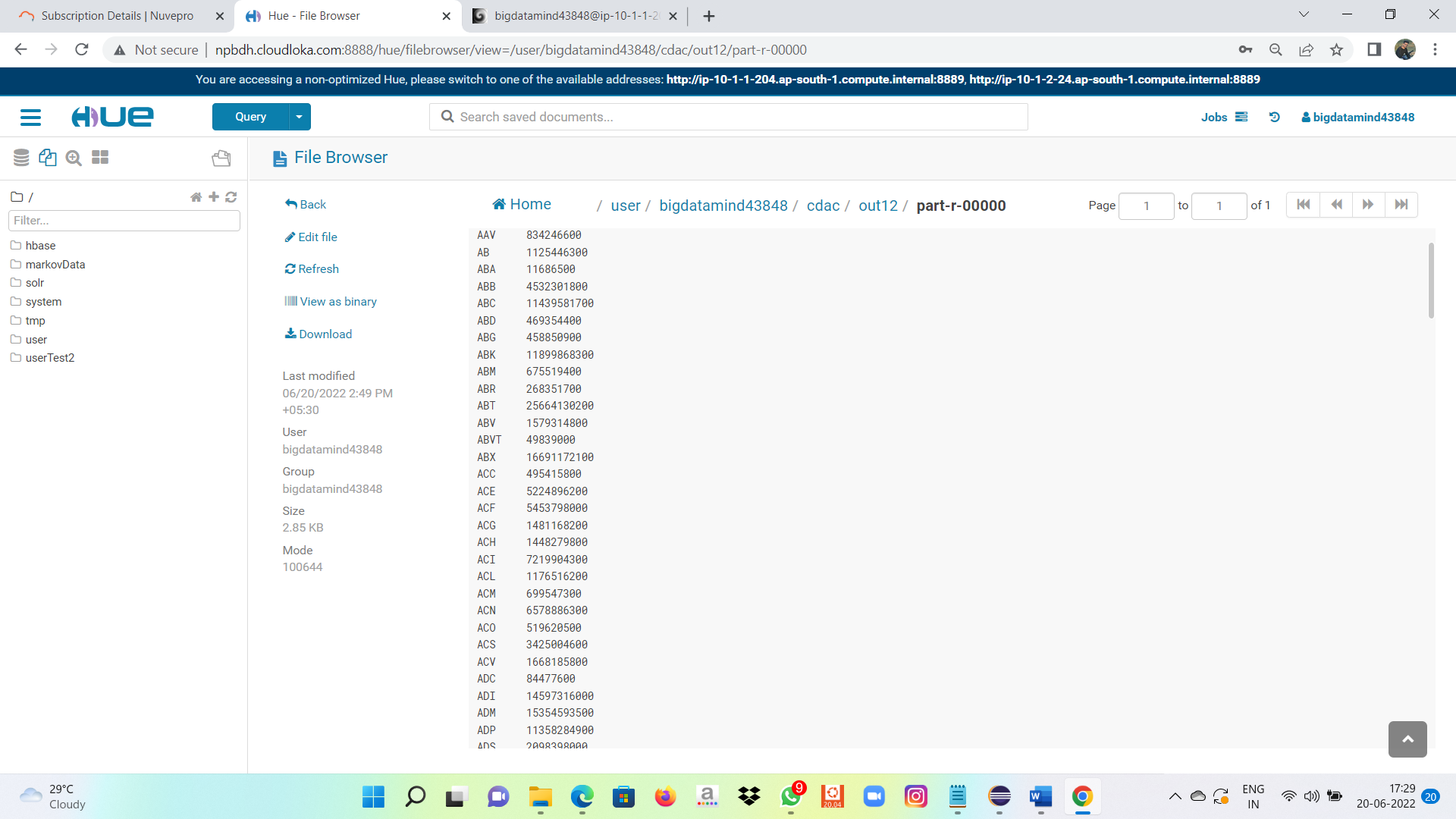
Que 1: map reduced:



Query:-

Hadoop jar myjar.jar cdac3/StockVolume cdac/NYSE.csv cdac/out12

Output:



Hive:-

1) -------------------------------

create table txnrecords(txnno INT, txndate STRING, custno INT, amount DOUBLE,

category STRING, product STRING, city STRING, state STRING, spendby STRING)

row format delimited

fields terminated by ','

stored as textfile;

reate table customer1(custno INT, firstname STRING, lastname STRING, age INT, profession STRING)

row format delimited

fields terminated by ','

stored as textfile;

create table txnrecsByCat(txnno INT, txndate STRING, custno INT, amount DOUBLE,

product STRING, city STRING, state STRING, spendby STRING)

partitioned by (category STRING)

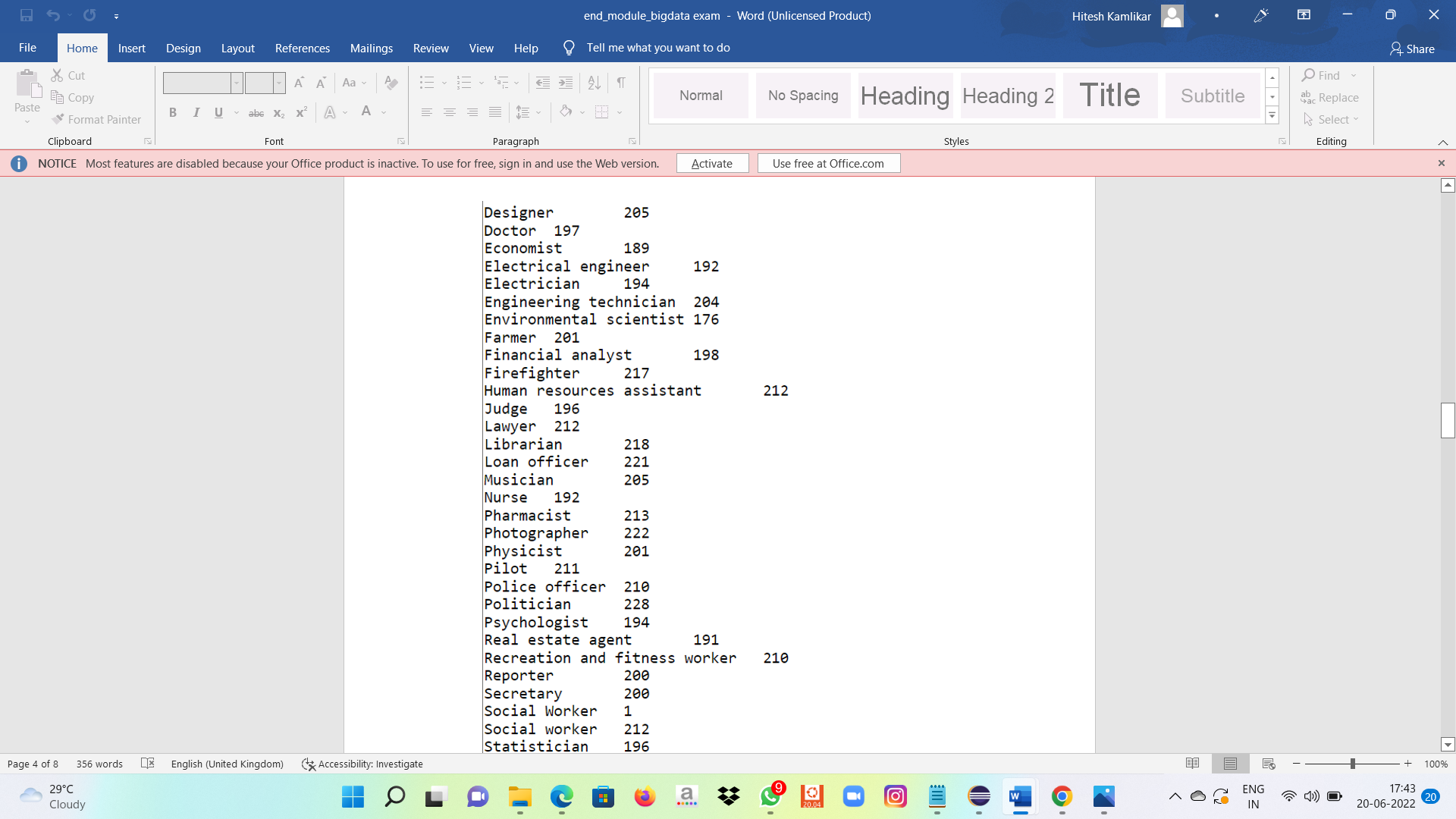
row format delimited

fields terminated by ','

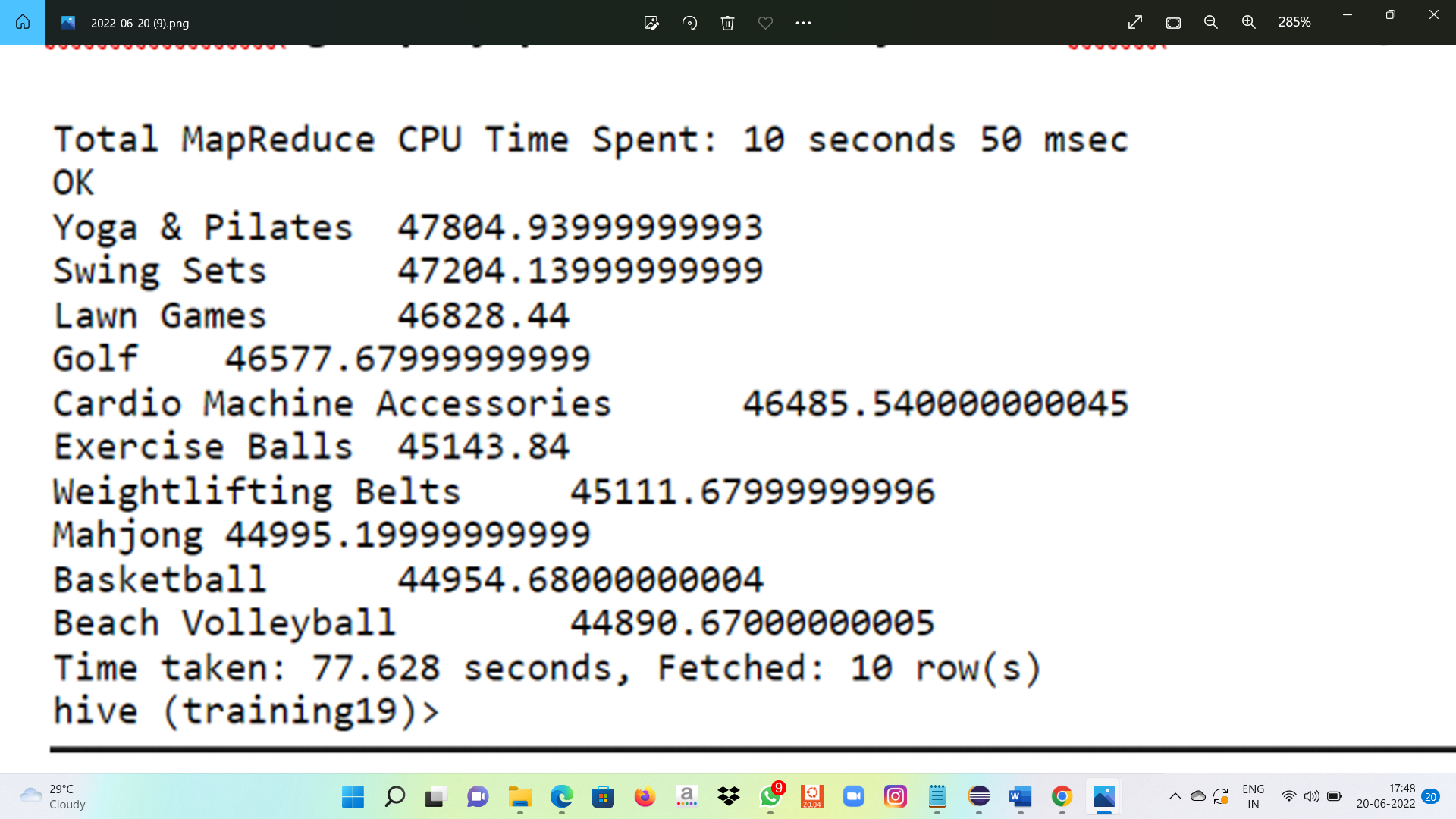
stored as textfile;

>> LOAD DATA LOCAL INPATH 'txns1.txt' OVERWRITE INTO TABLE txnrecords;

>> LOAD DATA LOCAL INPATH 'custs.txt' OVERWRITE INTO TABLE customer1;



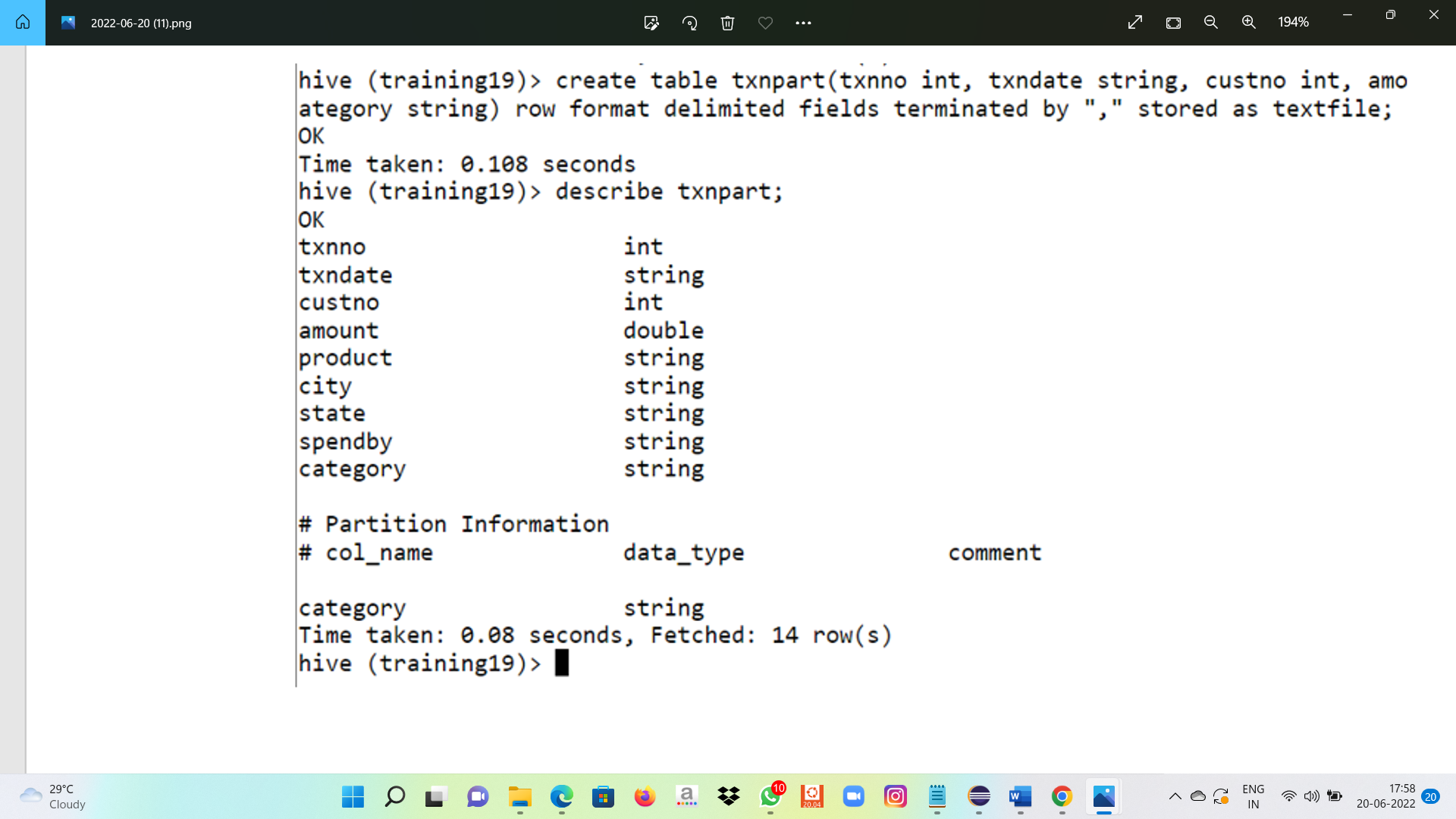
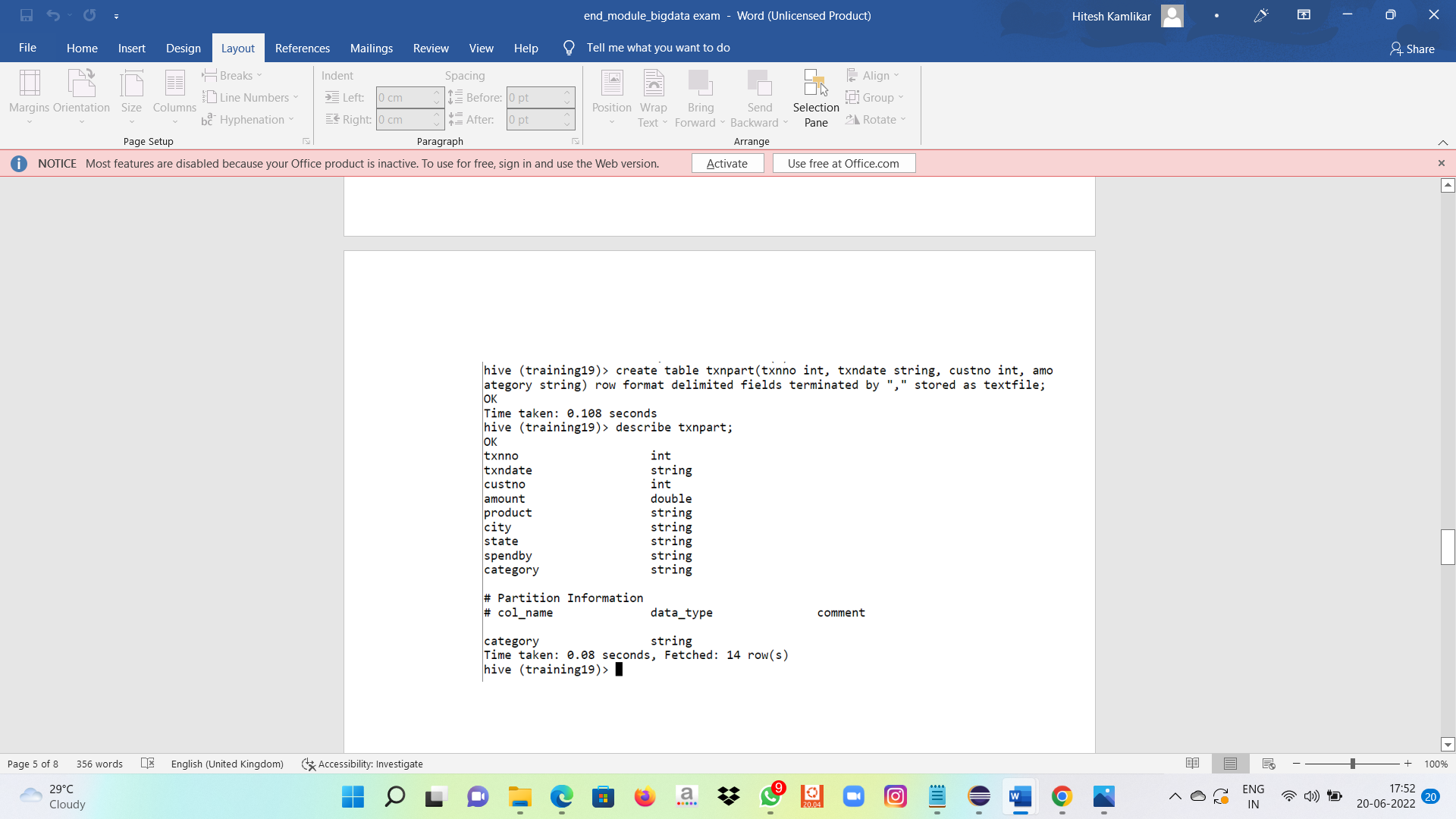
2)write a program to produce top 10 sales.



3) set hive.exec.dynamic.partition.mode=nonstrict;

set hive.exec.dynamic.partition=true;

INSERT OVERWRITE TABLE txnrecsByCat PARTITION(category) select txn.txnno, txn.txndate,txn.custno, txn.amount,txn.product,txn.city,txn.state, txn.spendby, txn.category from txnrecords txn DISTRIBUTE By category;



Pysparks:

1)

irlineRDD = sc.textFile("/user/bigdatamind43850/airlines.csv")

>>> airlineRDD.count()

85

>>> airlineRDD1 = airlineRDD.map(lambda a : a.encode("ascii","ignore"))

>>> header = airlineRDD1.first()

>>> airlineRDD2 = airlineRDD1.filter(lambda a : a != header)

>>> airlineRDD2.count()

84

>>> arrayRDD = airlineRDD2.map(lambda a : a.split(","))

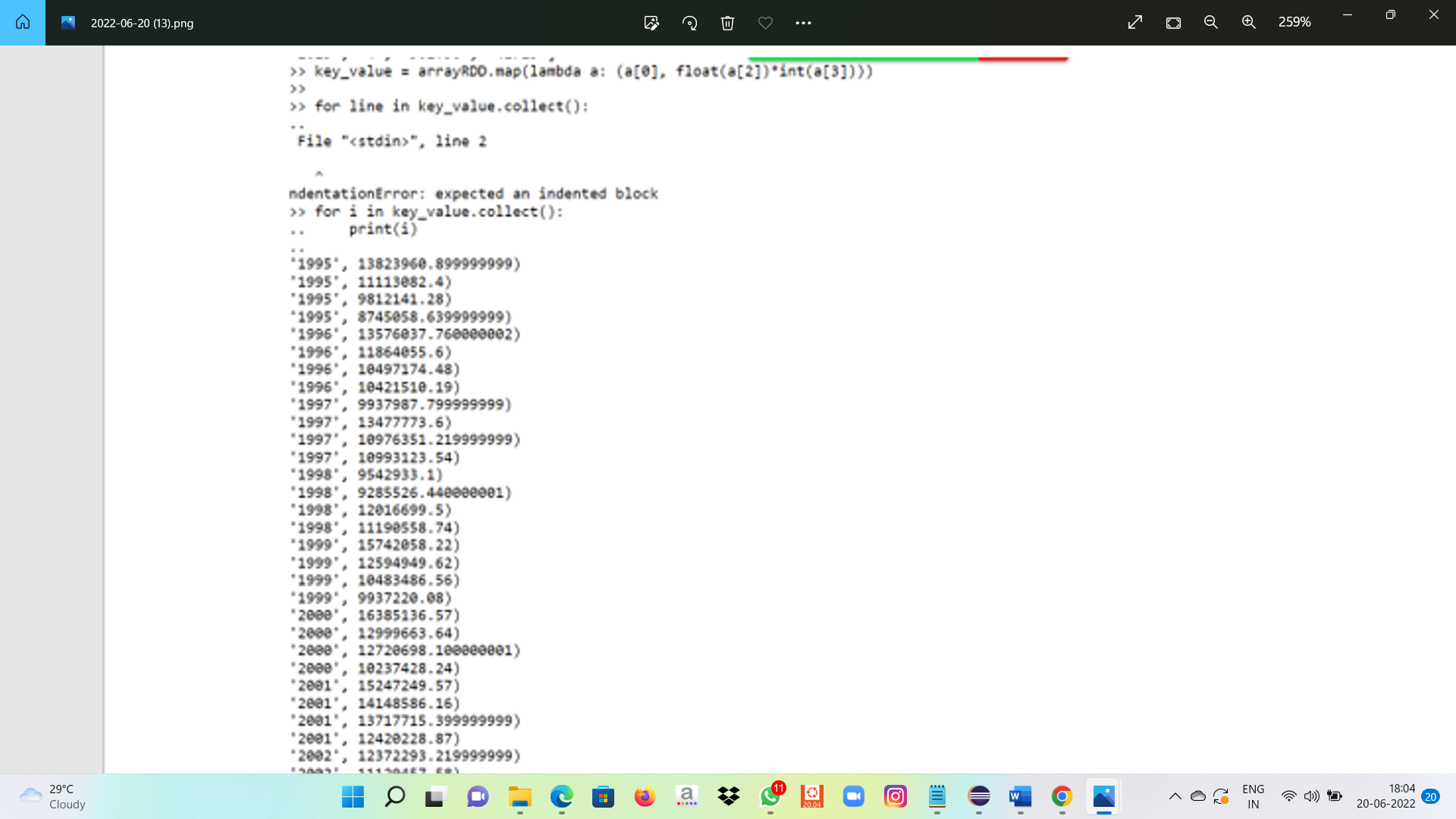
>>> keyvalue = arrayRDD.map(lambda a : (a[0],float(a[2])\*float(a[3])))

>>> for line in keyvalue.collect():

... print(line)

…

(‘2007”,176299)



2)

from pyspark.sql.types import StructType, IntegerType, DoubleType, LongType, StringType

schema2 = StructType().add("Year",StringType(),True).add("qtr",IntegerType(),True).add("revenue",DoubleType(),True).add("seats",LongType(),True)

airlinesDF = spark.read.format("csv").option("header","true").schema(schema2).load("/user/bigdatamind43846/Spark/airlines.csv")

