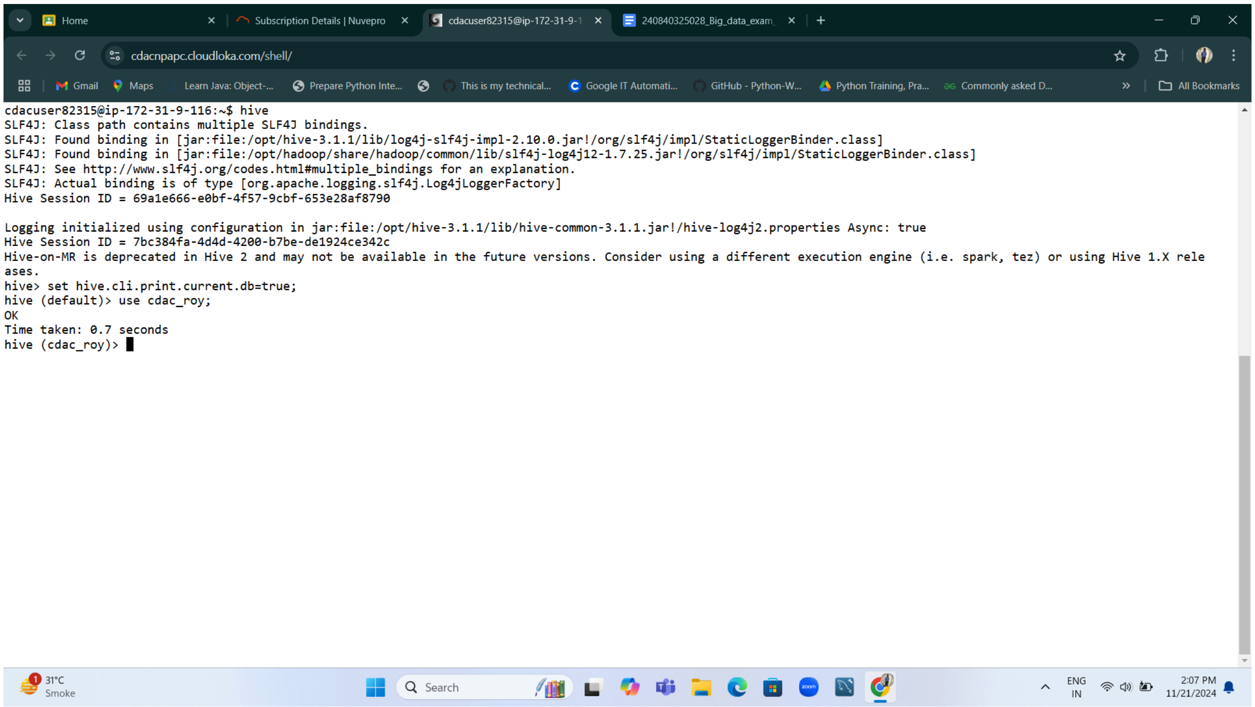
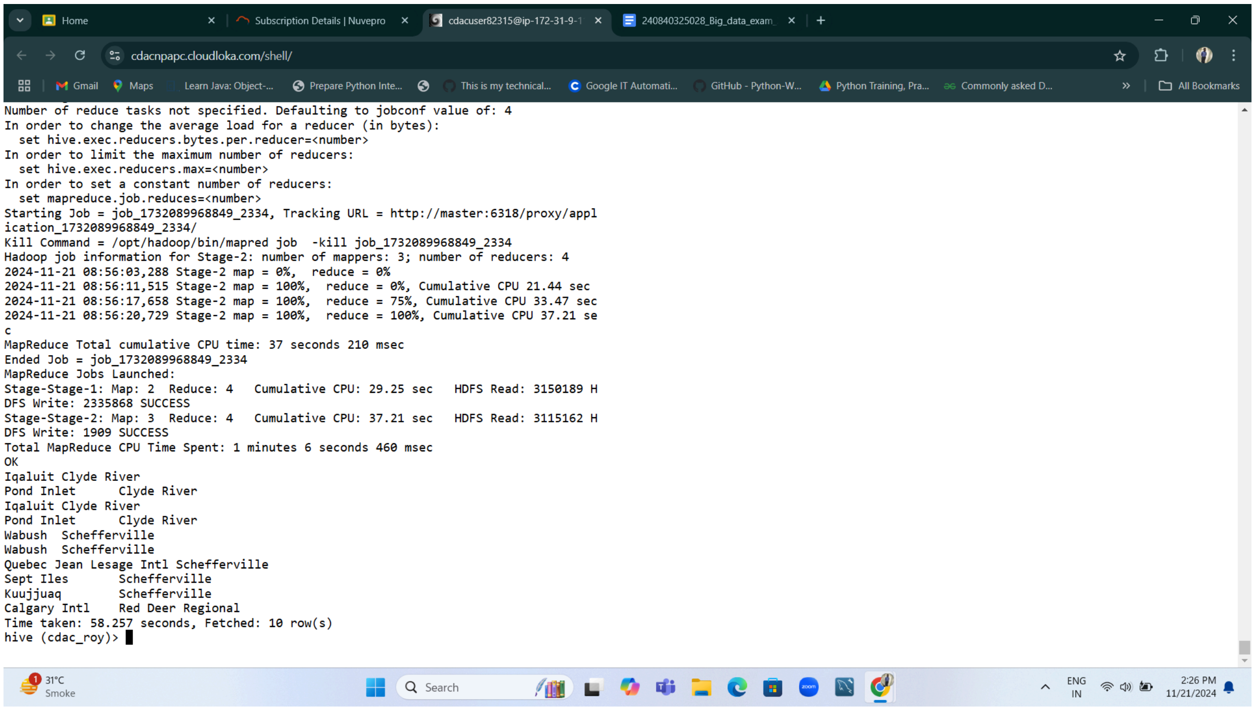
Big_data_exam: 240840325028

Q1.) 1.)



```
cdacuser82315@ip-172-31-9-116:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive-3.1.1/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 69a1e666-e0bf-4f57-9cbf-653e28af8790

Logging initialized using configuration in jar:file:/opt/hive-3.1.1/lib/hive-common-3.1.1.jar!/hive-log4j2.properties Async: true
Hive Session ID = 7bc384fa-4d4d-4200-b7be-de1924ce342c
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X rele
ases.
hive> set hive.cli.print.current.db=true;
hive (default)> use cdac_roy;
OK
Time taken: 0.7 seconds
hive (cdac_roy)> 
```

```
select a1.name,a2.name from airport a1 join routes r on
a1.airport_id = r.src_airport_id join airport a2 on
a2.airport_id = r.dest_airport_id limit 10;
```



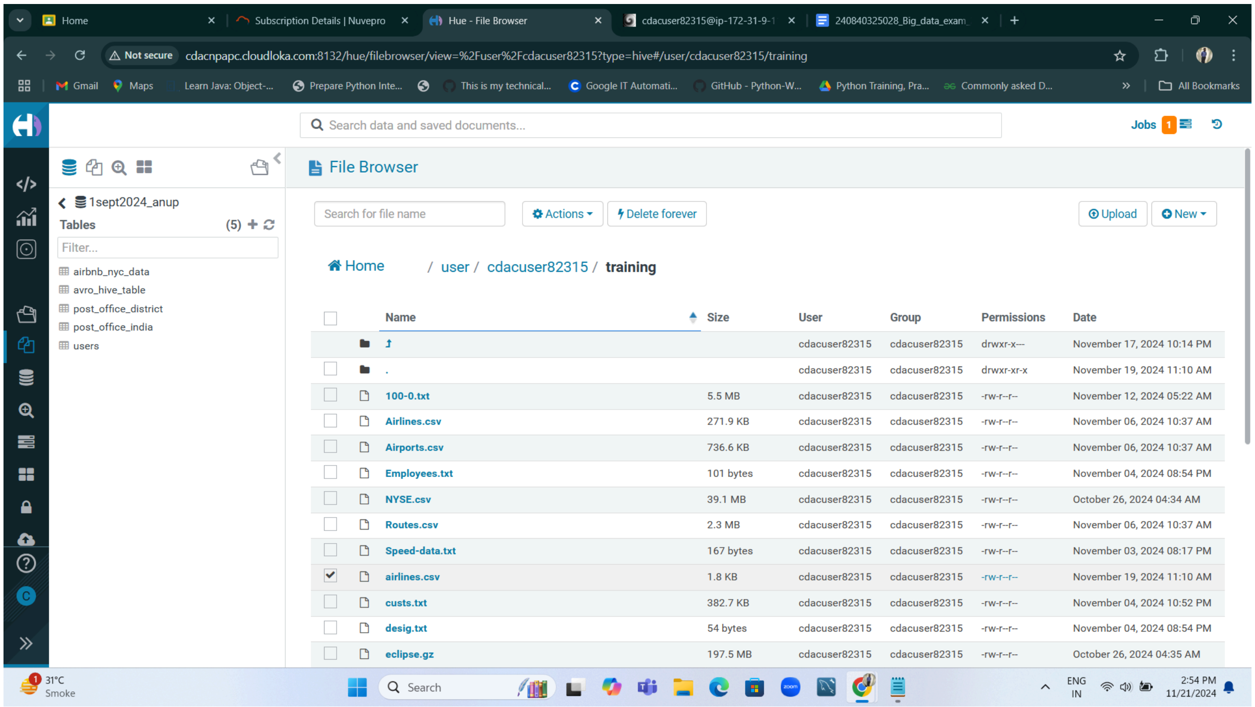
```
Number of reduce tasks not specified. Defaulting to jobconf value of: 4
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1732089968849_2334, Tracking URL = http://master:6318/proxy/appl
ication_1732089968849_2334/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_2334
Hadoop job information for Stage-2: number of mappers: 3; number of reducers: 4
2024-11-21 08:56:03,288 Stage-2 map = 0%,  reduce = 0%
2024-11-21 08:56:11,515 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 21.44 sec
2024-11-21 08:56:17,658 Stage-2 map = 100%,  reduce = 75%, Cumulative CPU 33.47 sec
2024-11-21 08:56:20,729 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 37.21 se
c
MapReduce Total cumulative CPU time: 37 seconds 210 msec
Ended Job = job_1732089968849_2334
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 4   Cumulative CPU: 29.25 sec   HDFS Read: 3150189 H
DFS Write: 2335868 SUCCESS
Stage-Stage-2: Map: 3  Reduce: 4   Cumulative CPU: 37.21 sec   HDFS Read: 3115162 H
DFS Write: 1909 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 6 seconds 460 msec
OK
Iqaluit Clyde River
Pond Inlet      Clyde River
Iqaluit Clyde River
Pond Inlet      Clyde River
Wabush  Schefferville
Wabush  Schefferville
Quebec Jean Lesage Intl Schefferville
Sept Iles       Schefferville
Kuujjuaq        Schefferville
Calgary Intl    Red Deer Regional
Time taken: 58.257 seconds, Fetched: 10 row(s)
hive (cdac_roy)> █
```

2.)

SPARK——---------

Q1.)

1.)

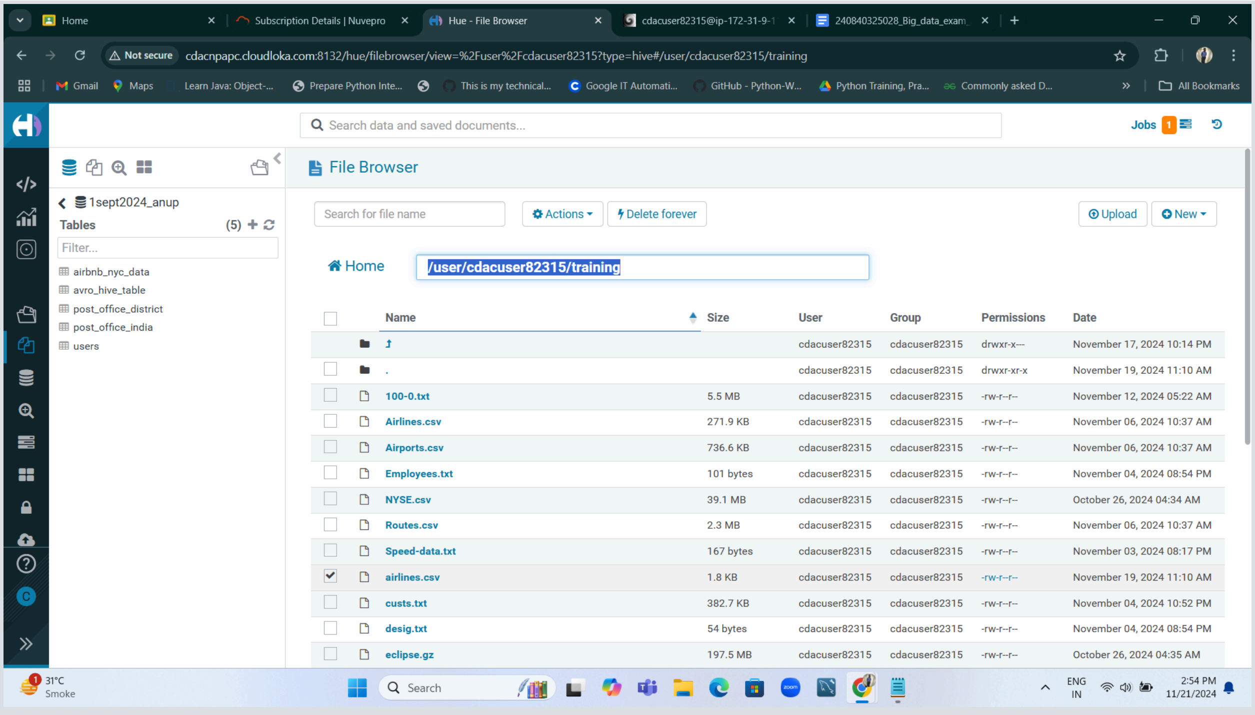


cdacnpapc.cloudloka.com:8132/hue/filebrowser/view=%2Fuser%2Fcdacuser82315?type=hive#/user/cdacuser82315/training

File Browser

Home / user / cdacuser82315 / training

Name	Size	User	Group	Permissions	Date
↥		cdacuser82315	cdacuser82315	drwxr-x---	November 17, 2024 10:14 PM
.		cdacuser82315	cdacuser82315	drwxr-xr-x	November 19, 2024 11:10 AM
100-0.txt	5.5 MB	cdacuser82315	cdacuser82315	-rw-r--r--	November 12, 2024 05:22 AM
Airlines.csv	271.9 KB	cdacuser82315	cdacuser82315	-rw-r--r--	November 06, 2024 10:37 AM
Airports.csv	736.6 KB	cdacuser82315	cdacuser82315	-rw-r--r--	November 06, 2024 10:37 AM
Employees.txt	101 bytes	cdacuser82315	cdacuser82315	-rw-r--r--	November 04, 2024 08:54 PM
NYSE.csv	39.1 MB	cdacuser82315	cdacuser82315	-rw-r--r--	October 26, 2024 04:34 AM
Routes.csv	2.3 MB	cdacuser82315	cdacuser82315	-rw-r--r--	November 06, 2024 10:37 AM
Speed-data.txt	167 bytes	cdacuser82315	cdacuser82315	-rw-r--r--	November 03, 2024 08:17 PM
☑ airlines.csv	1.8 KB	cdacuser82315	cdacuser82315	-rw-r--r--	November 19, 2024 11:10 AM
custs.txt	382.7 KB	cdacuser82315	cdacuser82315	-rw-r--r--	November 04, 2024 10:52 PM
desig.txt	54 bytes	cdacuser82315	cdacuser82315	-rw-r--r--	November 04, 2024 08:54 PM
eclipse.gz	197.5 MB	cdacuser82315	cdacuser82315	-rw-r--r--	October 26, 2024 04:35 AM

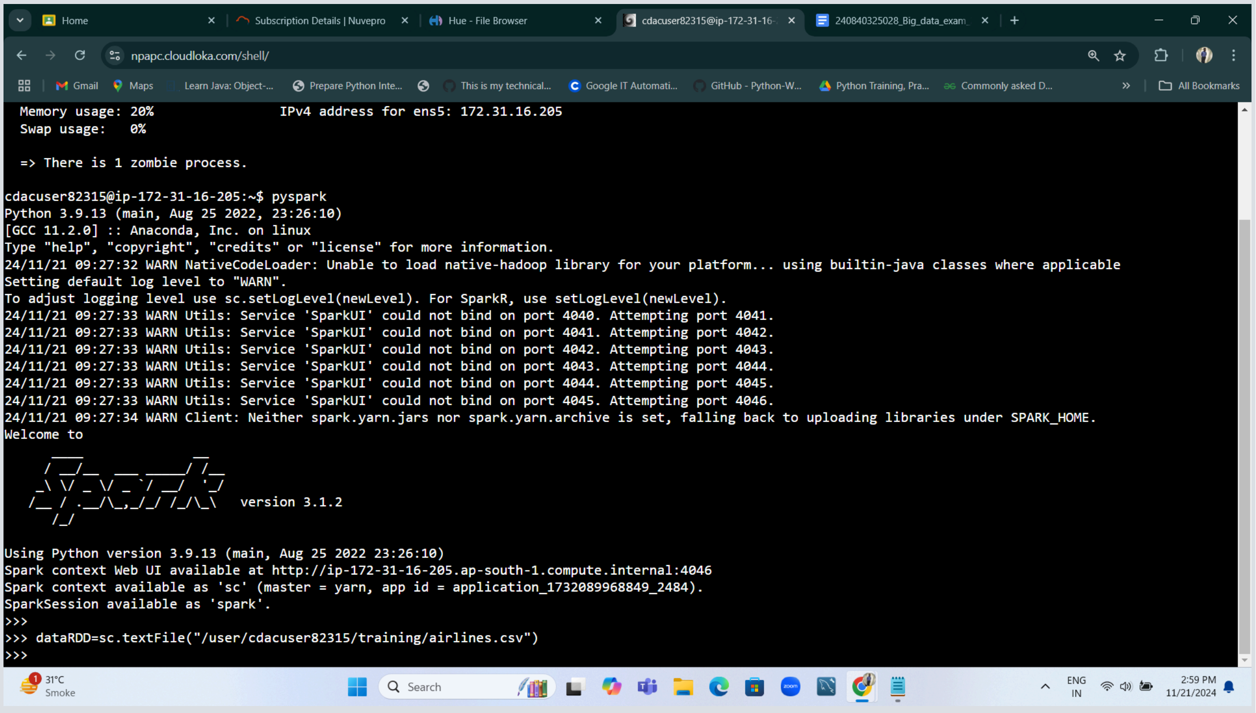


cdacnpapc.cloudloka.com:8132/hue/filebrowser/view=%2Fuser%2Fcdacuser82315?type=hive#/user/cdacuser82315/training

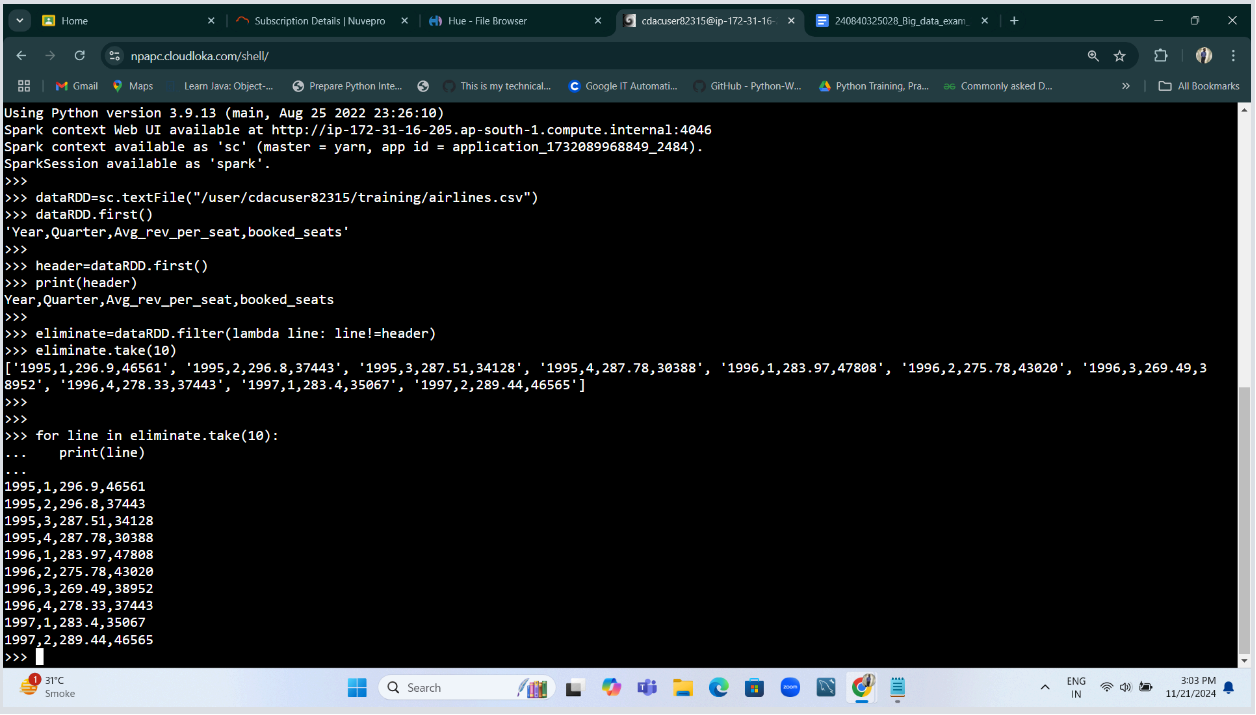
File Browser

Home /user/cdacuser82315/training

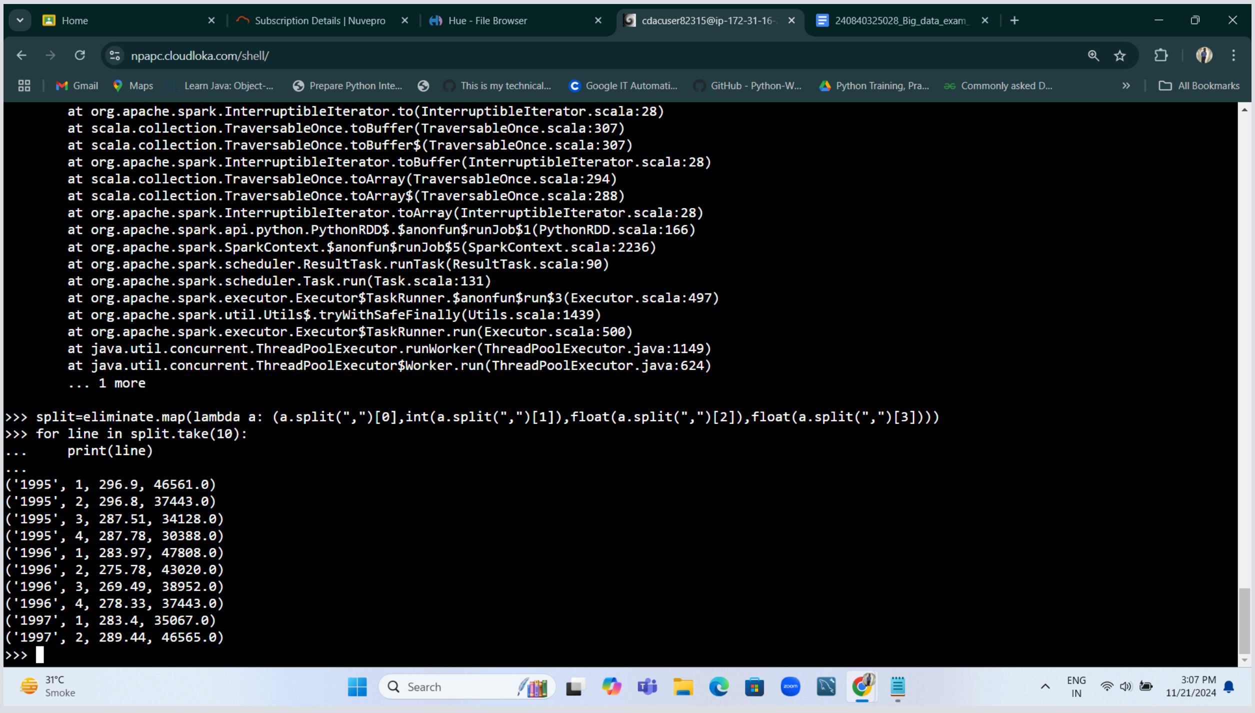
Name	Size	User	Group	Permissions	Date
↥		cdacuser82315	cdacuser82315	drwxr-x---	November 17, 2024 10:14 PM
.		cdacuser82315	cdacuser82315	drwxr-xr-x	November 19, 2024 11:10 AM
100-0.txt	5.5 MB	cdacuser82315	cdacuser82315	-rw-r--r--	November 12, 2024 05:22 AM
Airlines.csv	271.9 KB	cdacuser82315	cdacuser82315	-rw-r--r--	November 06, 2024 10:37 AM
Airports.csv	736.6 KB	cdacuser82315	cdacuser82315	-rw-r--r--	November 06, 2024 10:37 AM
Employees.txt	101 bytes	cdacuser82315	cdacuser82315	-rw-r--r--	November 04, 2024 08:54 PM
NYSE.csv	39.1 MB	cdacuser82315	cdacuser82315	-rw-r--r--	October 26, 2024 04:34 AM
Routes.csv	2.3 MB	cdacuser82315	cdacuser82315	-rw-r--r--	November 06, 2024 10:37 AM
Speed-data.txt	167 bytes	cdacuser82315	cdacuser82315	-rw-r--r--	November 03, 2024 08:17 PM
☑ airlines.csv	1.8 KB	cdacuser82315	cdacuser82315	-rw-r--r--	November 19, 2024 11:10 AM
custs.txt	382.7 KB	cdacuser82315	cdacuser82315	-rw-r--r--	November 04, 2024 10:52 PM
desig.txt	54 bytes	cdacuser82315	cdacuser82315	-rw-r--r--	November 04, 2024 08:54 PM
eclipse.gz	197.5 MB	cdacuser82315	cdacuser82315	-rw-r--r--	October 26, 2024 04:35 AM



```
  Memory usage: 20%               IPv4 address for ens5: 172.31.16.205
  Swap usage:   0%

  => There is 1 zombie process.

cdacuser82315@ip-172-31-16-205:~$ pyspark
Python 3.9.13 (main, Aug 25 2022, 23:26:10)
[GCC 11.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
24/11/21 09:27:32 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/11/21 09:27:33 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
24/11/21 09:27:33 WARN Utils: Service 'SparkUI' could not bind on port 4041. Attempting port 4042.
24/11/21 09:27:33 WARN Utils: Service 'SparkUI' could not bind on port 4042. Attempting port 4043.
24/11/21 09:27:33 WARN Utils: Service 'SparkUI' could not bind on port 4043. Attempting port 4044.
24/11/21 09:27:33 WARN Utils: Service 'SparkUI' could not bind on port 4044. Attempting port 4045.
24/11/21 09:27:33 WARN Utils: Service 'SparkUI' could not bind on port 4045. Attempting port 4046.
24/11/21 09:27:34 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.2
      /_/

Using Python version 3.9.13 (main, Aug 25 2022 23:26:10)
Spark context Web UI available at http://ip-172-31-16-205.ap-south-1.compute.internal:4046
Spark context available as 'sc' (master = yarn, app id = application_1732089968849_2484).
SparkSession available as 'spark'.
>>>
>>> dataRDD=sc.textFile("/user/cdacuser82315/training/airlines.csv")
>>>
```

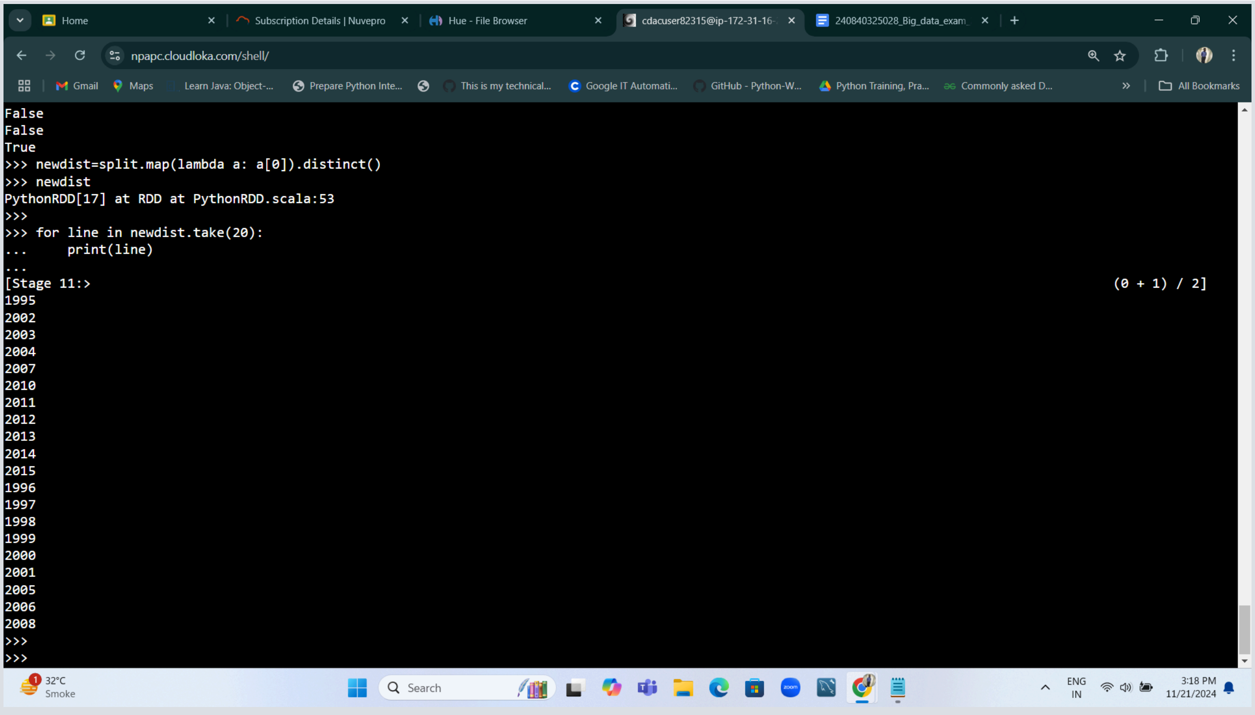


```
Using Python version 3.9.13 (main, Aug 25 2022 23:26:10)
Spark context Web UI available at http://ip-172-31-16-205.ap-south-1.compute.internal:4046
Spark context available as 'sc' (master = yarn, app id = application_1732089968849_2484).
SparkSession available as 'spark'.
>>>
>>> dataRDD=sc.textFile("/user/cdacuser82315/training/airlines.csv")
>>> dataRDD.first()
'Year,Quarter,Avg_rev_per_seat,booked_seats'
>>>
>>> header=dataRDD.first()
>>> print(header)
Year,Quarter,Avg_rev_per_seat,booked_seats
>>>
>>> eliminate=dataRDD.filter(lambda line: line!=header)
>>> eliminate.take(10)
['1995,1,296.9,46561', '1995,2,296.8,37443', '1995,3,287.51,34128', '1995,4,287.78,30388', '1996,1,283.97,47808', '1996,2,275.78,43020', '1996,3,269.49,3
8952', '1996,4,278.33,37443', '1997,1,283.4,35067', '1997,2,289.44,46565']
>>>
>>>
>>> for line in eliminate.take(10):
...    print(line)
...
1995,1,296.9,46561
1995,2,296.8,37443
1995,3,287.51,34128
1995,4,287.78,30388
1996,1,283.97,47808
1996,2,275.78,43020
1996,3,269.49,38952
1996,4,278.33,37443
1997,1,283.4,35067
1997,2,289.44,46565
>>>
```



```
        at org.apache.spark.InterruptibleIterator.to(InterruptibleIterator.scala:28)
        at scala.collection.TraversableOnce.toBuffer(TraversableOnce.scala:307)
        at scala.collection.TraversableOnce.toBuffer$(TraversableOnce.scala:307)
        at org.apache.spark.InterruptibleIterator.toBuffer(InterruptibleIterator.scala:28)
        at scala.collection.TraversableOnce.toArray(TraversableOnce.scala:294)
        at scala.collection.TraversableOnce.toArray$(TraversableOnce.scala:288)
        at org.apache.spark.InterruptibleIterator.toArray(InterruptibleIterator.scala:28)
        at org.apache.spark.api.python.PythonRDD$.$anonfun$runJob$1(PythonRDD.scala:166)
        at org.apache.spark.SparkContext.$anonfun$runJob$5(SparkContext.scala:2236)
        at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:90)
        at org.apache.spark.scheduler.Task.run(Task.scala:131)
        at org.apache.spark.executor.Executor$TaskRunner.$anonfun$run$3(Executor.scala:497)
        at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:1439)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:500)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        ... 1 more

>>> split=eliminate.map(lambda a: (a.split(",")[0],int(a.split(",")[1]),float(a.split(",")[2]),float(a.split(",")[3])))
>>> for line in split.take(10):
...     print(line)
...
('1995', 1, 296.9, 46561.0)
('1995', 2, 296.8, 37443.0)
('1995', 3, 287.51, 34128.0)
('1995', 4, 287.78, 30388.0)
('1996', 1, 283.97, 47808.0)
('1996', 2, 275.78, 43020.0)
('1996', 3, 269.49, 38952.0)
('1996', 4, 278.33, 37443.0)
('1997', 1, 283.4, 35067.0)
('1997', 2, 289.44, 46565.0)
>>>
```

2.)

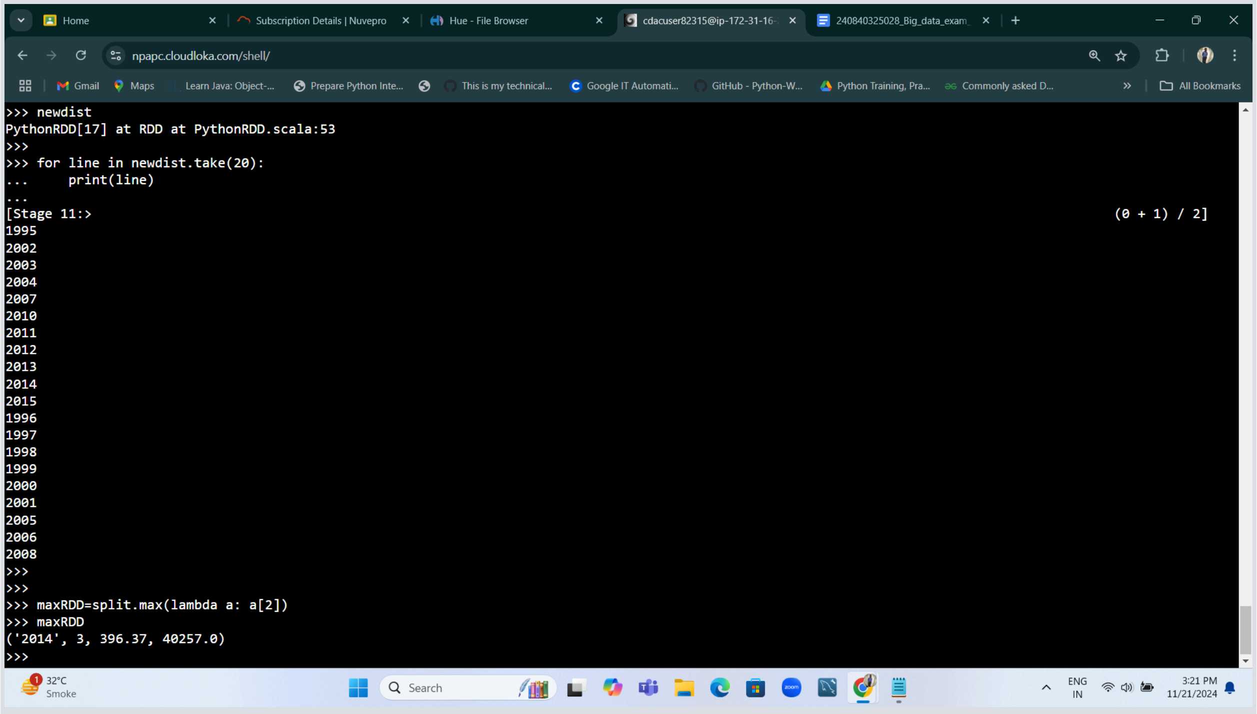


```
False
False
True
>>> newdist=split.map(lambda a: a[0]).distinct()
>>> newdist
PythonRDD[17] at RDD at PythonRDD.scala:53
>>>
>>> for line in newdist.take(20):
...     print(line)
...
[Stage 11:>                                                                 (0 + 1) / 2]
1995
2002
2003
2004
2007
2010
2011
2012
2013
2014
2015
1996
1997
1998
1999
2000
2001
2005
2006
2008
>>>
>>>
```

Question 2.)

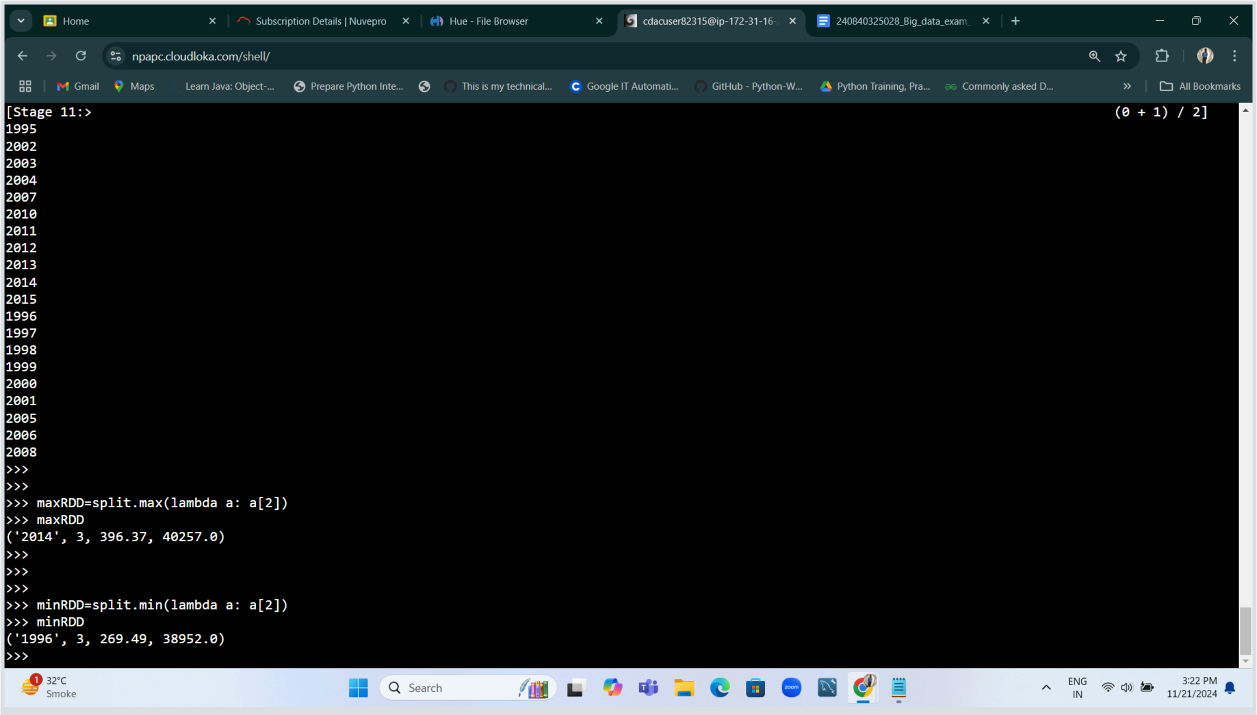
1.)

```
maxRDD=split.max(lambda a: a[2])
maxRDD
```



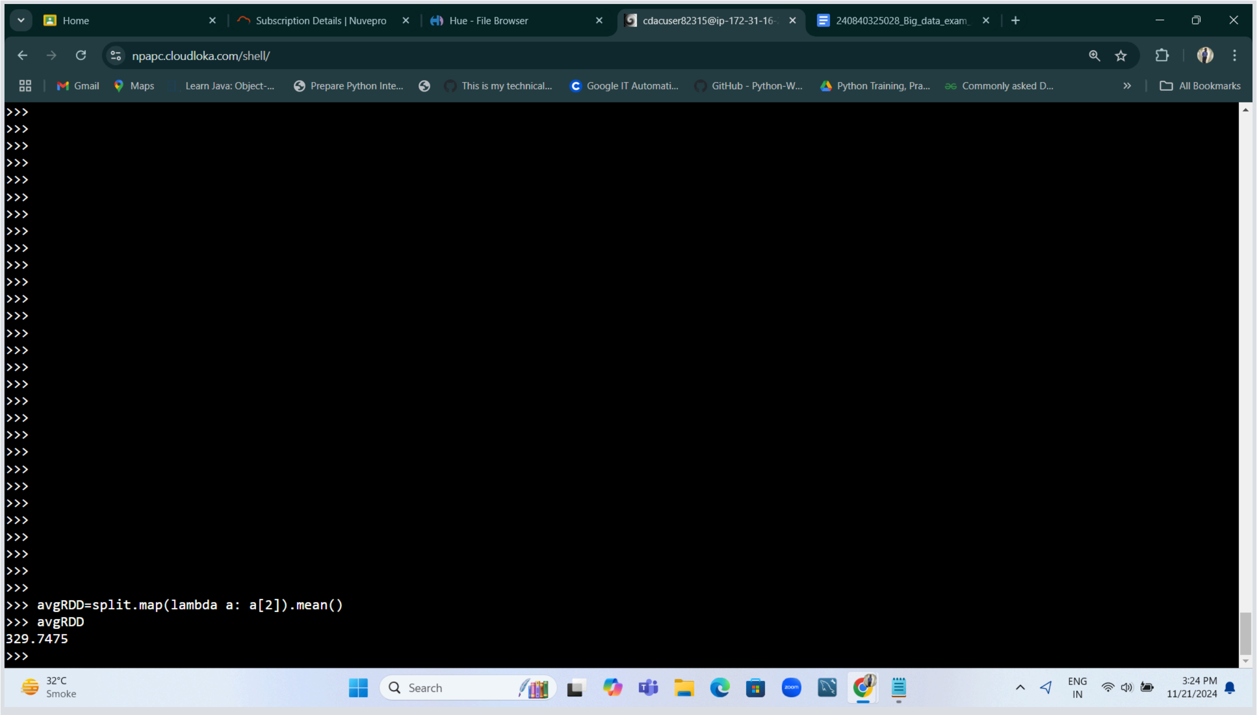
```
>>> newdist
PythonRDD[17] at RDD at PythonRDD.scala:53
>>>
>>> for line in newdist.take(20):
...     print(line)
...
[Stage 11:>                                                                     (0 + 1) / 2]
1995
2002
2003
2004
2007
2010
2011
2012
2013
2014
2015
1996
1997
1998
1999
2000
2001
2005
2006
2008
>>>
>>>
>>> maxRDD=split.max(lambda a: a[2])
>>> maxRDD
('2014', 3, 396.37, 40257.0)
>>>
```

```
minRDD=split.min(lambda a: a[2])
minRDD
```



```
[Stage 11:>                                                          (0 + 1) / 2]
1995
2002
2003
2004
2007
2010
2011
2012
2013
2014
2015
1996
1997
1998
1999
2000
2001
2005
2006
2008
>>>
>>>
>>> maxRDD=split.max(lambda a: a[2])
>>> maxRDD
('2014', 3, 396.37, 40257.0)
>>>
>>>
>>>
>>> minRDD=split.min(lambda a: a[2])
>>> minRDD
('1996', 3, 269.49, 38952.0)
>>>
```

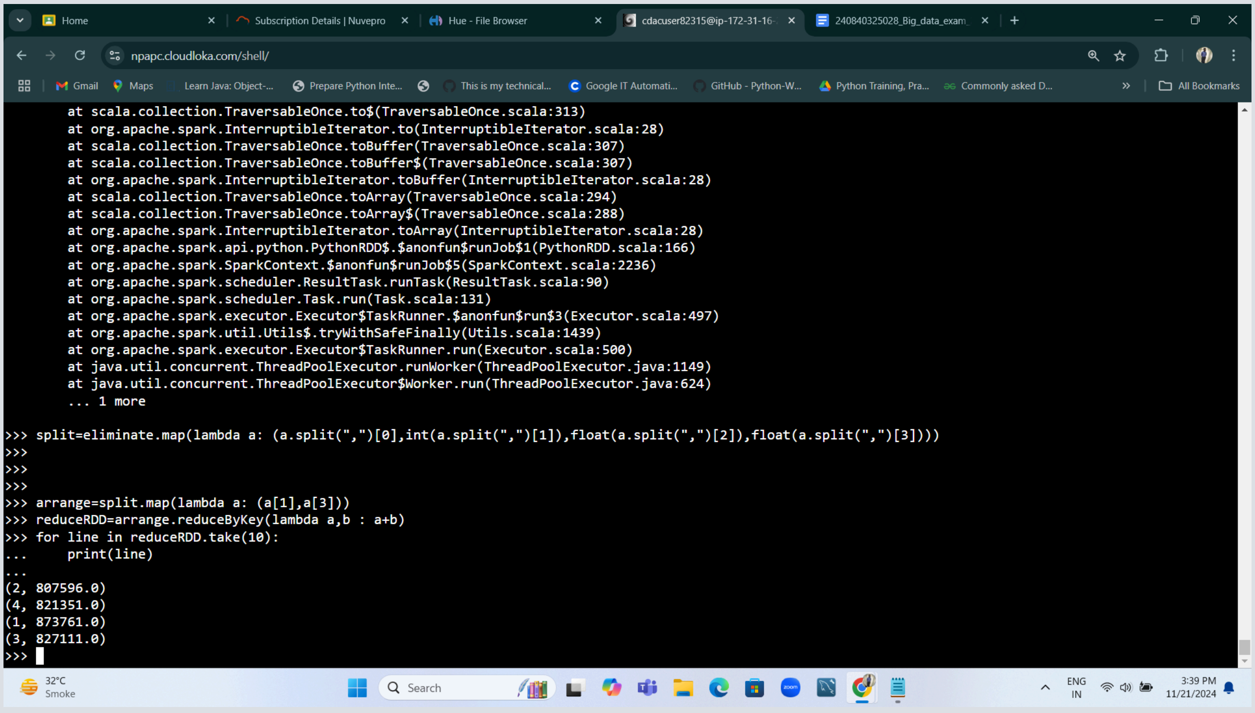
```
avgRDD=split.map(lambda a: a[2]).mean()
avgRDD
329.7475
```



```
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> avgRDD=split.map(lambda a: a[2]).mean()
>>> avgRDD
329.7475
>>>
```

3.)

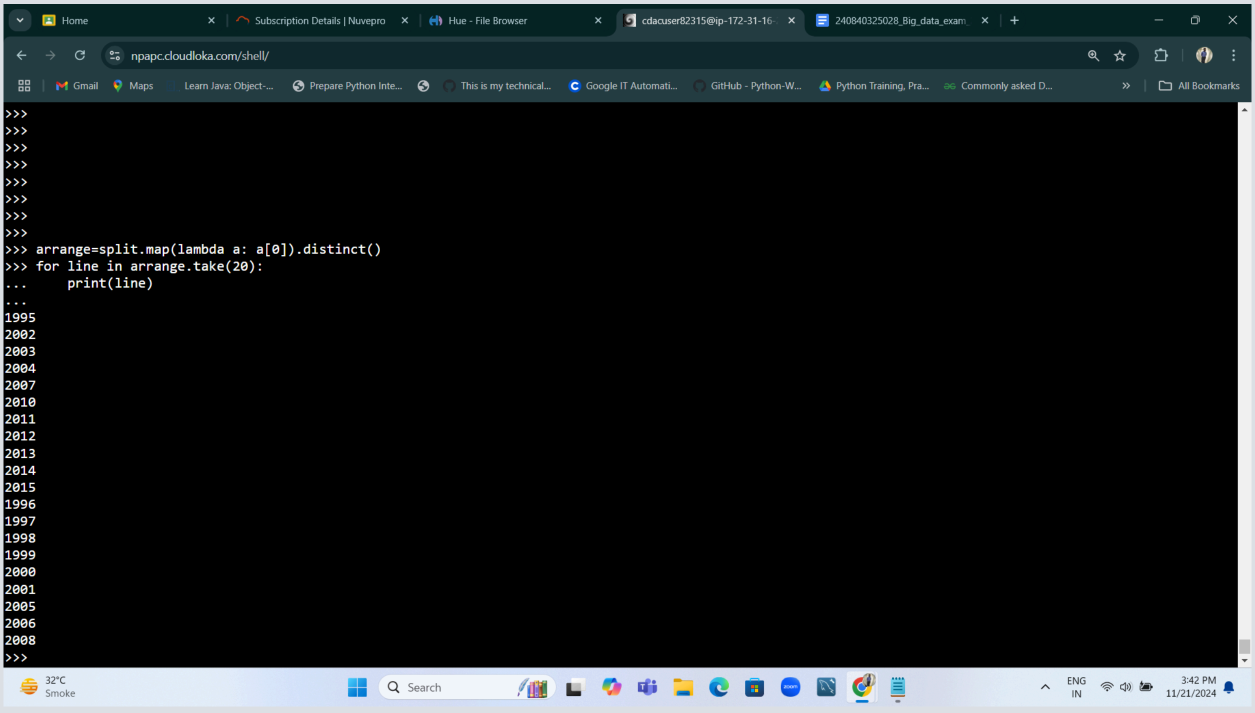
```
arrange=split.map(lambda a:  (a[1],a[3]))
>>> reduceRDD=arrange.reduceByKey(lambda a,b : a+b)
>>> for line in reduceRDD.take(10):
...       print(line)
```



```
        at scala.collection.TraversableOnce.to$(TraversableOnce.scala:313)
        at org.apache.spark.InterruptibleIterator.to(InterruptibleIterator.scala:28)
        at scala.collection.TraversableOnce.toBuffer(TraversableOnce.scala:307)
        at scala.collection.TraversableOnce.toBuffer$(TraversableOnce.scala:307)
        at org.apache.spark.InterruptibleIterator.toBuffer(InterruptibleIterator.scala:28)
        at scala.collection.TraversableOnce.toArray(TraversableOnce.scala:294)
        at scala.collection.TraversableOnce.toArray$(TraversableOnce.scala:288)
        at org.apache.spark.InterruptibleIterator.toArray(InterruptibleIterator.scala:28)
        at org.apache.spark.api.python.PythonRDD$.$anonfun$runJob$1(PythonRDD.scala:166)
        at org.apache.spark.SparkContext.$anonfun$runJob$5(SparkContext.scala:2236)
        at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:90)
        at org.apache.spark.scheduler.Task.run(Task.scala:131)
        at org.apache.spark.executor.Executor$TaskRunner.$anonfun$run$3(Executor.scala:497)
        at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:1439)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:500)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        ... 1 more

>>> split=eliminate.map(lambda a: (a.split(",")[0],int(a.split(",")[1]),float(a.split(",")[2]),float(a.split(",")[3])))
>>>
>>>
>>>
>>> arrange=split.map(lambda a: (a[1],a[3]))
>>> reduceRDD=arrange.reduceByKey(lambda a,b : a+b)
>>> for line in reduceRDD.take(10):
...     print(line)
...
(2, 807596.0)
(4, 821351.0)
(1, 873761.0)
(3, 827111.0)
>>> 
```

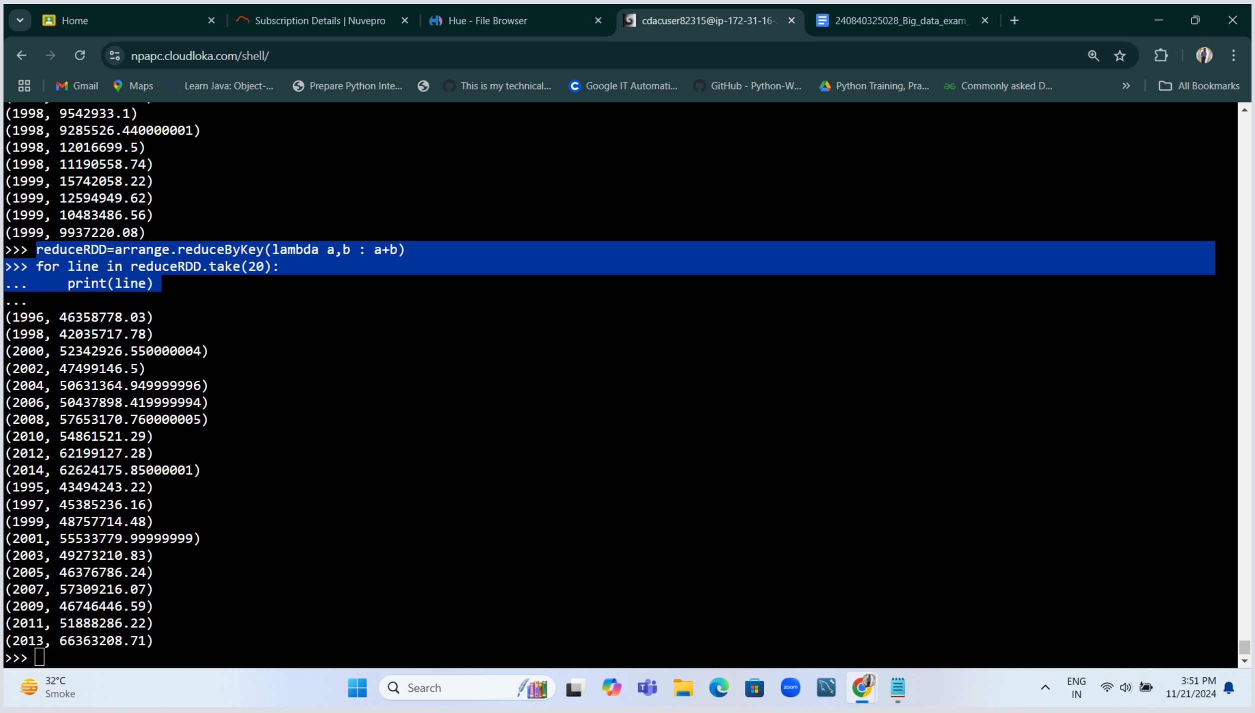
4.)

```
arrange=split.map(lambda a: a[0]).distinct()
>>> for line in arrange.take(20):
...      print(line)
```



```
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> arrange=split.map(lambda a: a[0]).distinct()
>>> for line in arrange.take(20):
...     print(line)
...
1995
2002
2003
2004
2007
2010
2011
2012
2013
2014
2015
1996
1997
1998
1999
2000
2001
2005
2006
2008
>>>
```

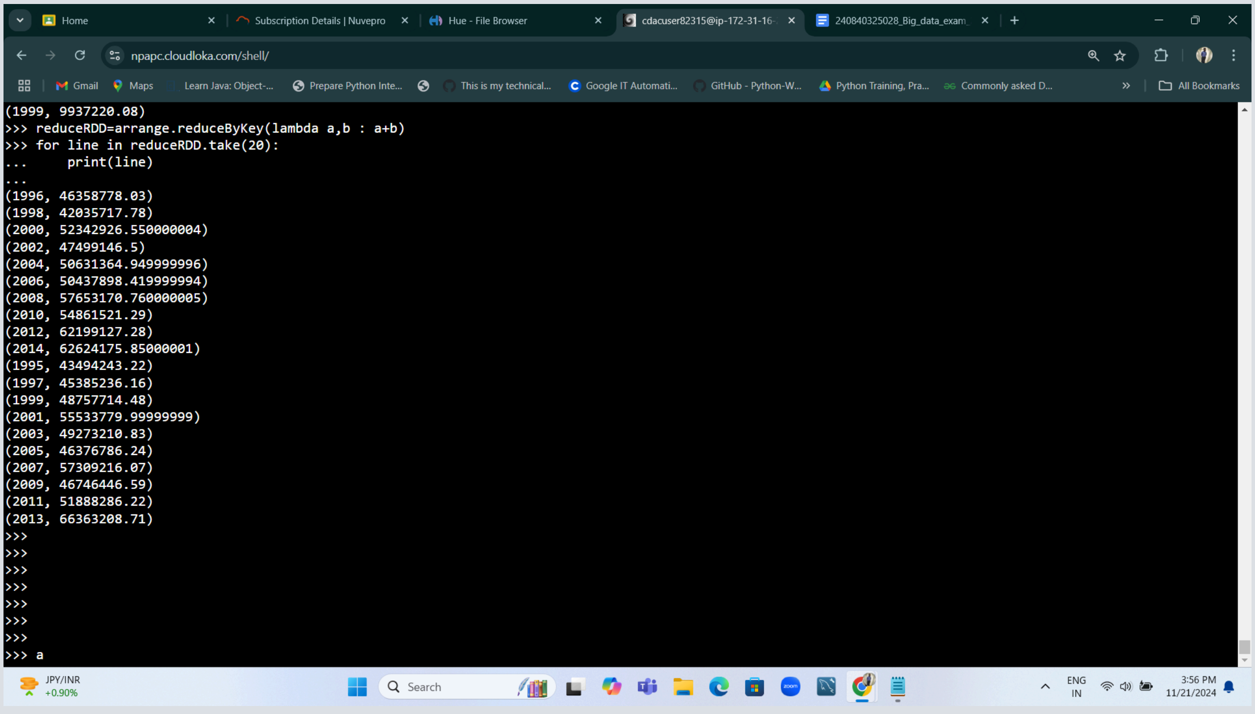
5.)
```
reduceRDD=arrange.reduceByKey(lambda a,b : a+b)
>>> for line in reduceRDD.take(20):
...       print(line)
```



```
(1998, 9542933.1)
(1998, 9285526.440000001)
(1998, 12016699.5)
(1998, 11190558.74)
(1999, 15742058.22)
(1999, 12594949.62)
(1999, 10483486.56)
(1999, 9937220.08)
>>> reduceRDD=arrange.reduceByKey(lambda a,b : a+b)
>>> for line in reduceRDD.take(20):
...     print(line)
...
(1996, 46358778.03)
(1998, 42035717.78)
(2000, 52342926.550000004)
(2002, 47499146.5)
(2004, 50631364.949999996)
(2006, 50437898.419999994)
(2008, 57653170.760000005)
(2010, 54861521.29)
(2012, 62199127.28)
(2014, 62624175.85000001)
(1995, 43494243.22)
(1997, 45385236.16)
(1999, 48757714.48)
(2001, 55533779.99999999)
(2003, 49273210.83)
(2005, 46376786.24)
(2007, 57309216.07)
(2009, 46746446.59)
(2011, 51888286.22)
(2013, 66363208.71)
>>> 
```

2.) groupy(‘year’,’quarter’).agg(count(col(avg_per_seat)>$290)

→Screen stuck written the query



```
(1999, 9937220.08)
>>> reduceRDD=arrange.reduceByKey(lambda a,b : a+b)
>>> for line in reduceRDD.take(20):
...     print(line)
...
(1996, 46358778.03)
(1998, 42035717.78)
(2000, 52342926.550000004)
(2002, 47499146.5)
(2004, 50631364.949999996)
(2006, 50437898.419999994)
(2008, 57653170.760000005)
(2010, 54861521.29)
(2012, 62199127.28)
(2014, 62624175.85000001)
(1995, 43494243.22)
(1997, 45385236.16)
(1999, 48757714.48)
(2001, 55533779.99999999)
(2003, 49273210.83)
(2005, 46376786.24)
(2007, 57309216.07)
(2009, 46746446.59)
(2011, 51888286.22)
(2013, 66363208.71)
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> a
```