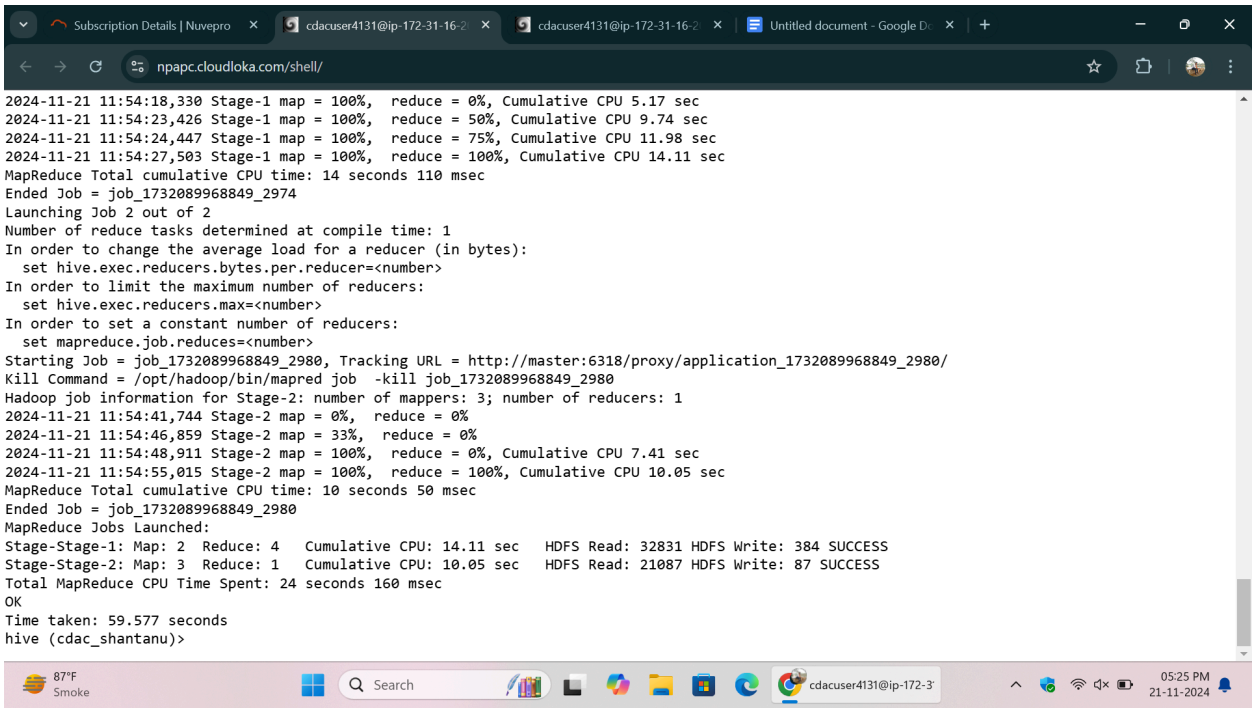
BIG DATA EXAM

I.HIVE

Q.1)

1) select airport.name from airport join routes on airport.iata=routes.src_airport_iata join routes r on airport.iata=r.dest_airport_iata where airport.iata = routes.src_airport_iata and airport.iata != r.dest_airport_iata limit 10;



```
2024-11-21 11:54:18,330 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.17 sec
2024-11-21 11:54:23,426 Stage-1 map = 100%,  reduce = 50%, Cumulative CPU 9.74 sec
2024-11-21 11:54:24,447 Stage-1 map = 100%,  reduce = 75%, Cumulative CPU 11.98 sec
2024-11-21 11:54:27,503 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 14.11 sec
MapReduce Total cumulative CPU time: 14 seconds 110 msec
Ended Job = job_1732089968849_2974
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
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_2980, Tracking URL = http://master:6318/proxy/application_1732089968849_2980/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_2980
Hadoop job information for Stage-2: number of mappers: 3; number of reducers: 1
2024-11-21 11:54:41,744 Stage-2 map = 0%,  reduce = 0%
2024-11-21 11:54:46,859 Stage-2 map = 33%,  reduce = 0%
2024-11-21 11:54:48,911 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 7.41 sec
2024-11-21 11:54:55,015 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 10.05 sec
MapReduce Total cumulative CPU time: 10 seconds 50 msec
Ended Job = job_1732089968849_2980
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 4   Cumulative CPU: 14.11 sec   HDFS Read: 32831 HDFS Write: 384 SUCCESS
Stage-Stage-2: Map: 3  Reduce: 1   Cumulative CPU: 10.05 sec   HDFS Read: 21087 HDFS Write: 87 SUCCESS
Total MapReduce CPU Time Spent: 24 seconds 160 msec
OK
Time taken: 59.577 seconds
hive (cdac_shantanu)>
```

3) (cdac_shantanu)> select count(distinct(equipment)) from routes;

```
 777
100 318
100 319 ER4
Time taken: 30.671 seconds, Fetched: 10 row(s)
hive (cdac_shantanu)> select count(distinct(equipment)) from routes;
Query ID = cdacuser4131_20241121121133_af7af30a-3313-41cb-8154-564c59b41776
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
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_3030, Tracking URL = http://master:6318/proxy/application_1732089968849_3030/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_3030
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-11-21 12:11:44,623 Stage-1 map = 0%,  reduce = 0%
2024-11-21 12:11:52,781 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.15 sec
2024-11-21 12:11:58,903 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 8.09 sec
MapReduce Total cumulative CPU time: 8 seconds 90 msec
Ended Job = job_1732089968849_3030
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 8.09 sec   HDFS Read: 2385325 HDFS Write: 104 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 90 msec
OK
3946
Time taken: 27.555 seconds, Fetched: 1 row(s)
hive (cdac_shantanu)>
```

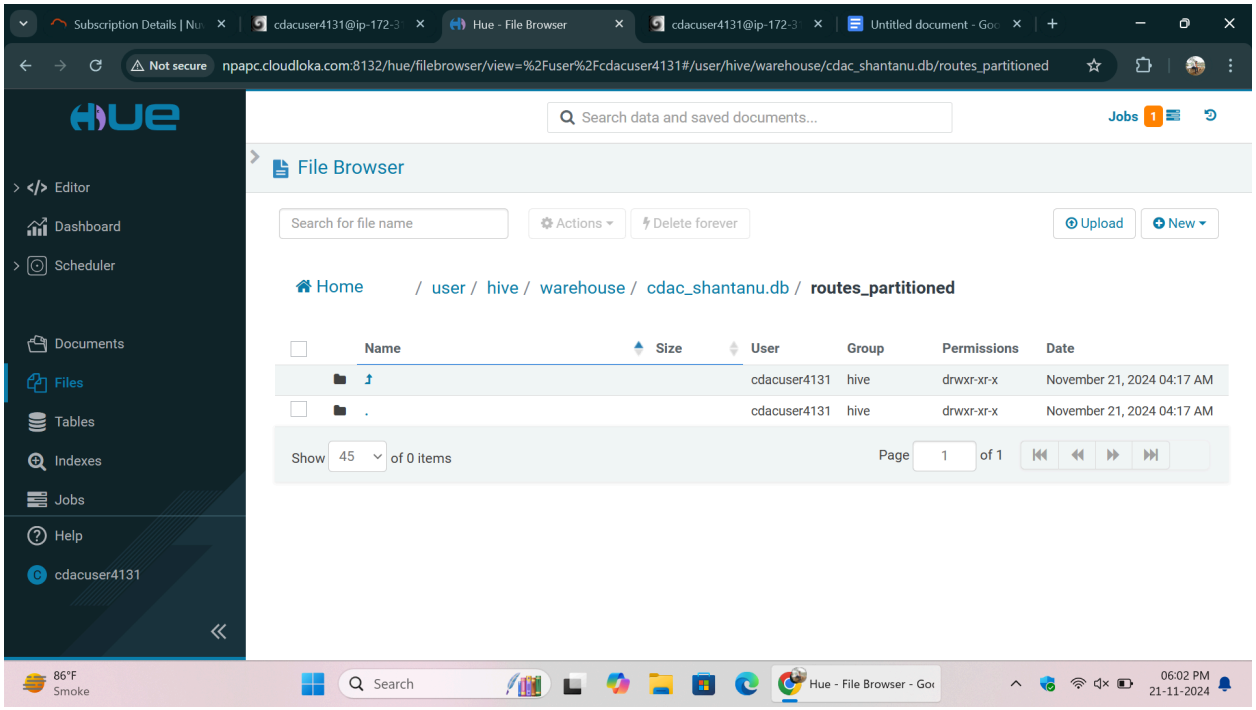
Q.2)

create table routes_partitioned(airline_iata string, airline_id int, src_airport_iata string, src_airport_id int, dest_airport_id int, codeshare string, stops int, equipment string) partitioned by(dest_airport_iata string) row format delimited fields terminated by ',' stored as textfile;

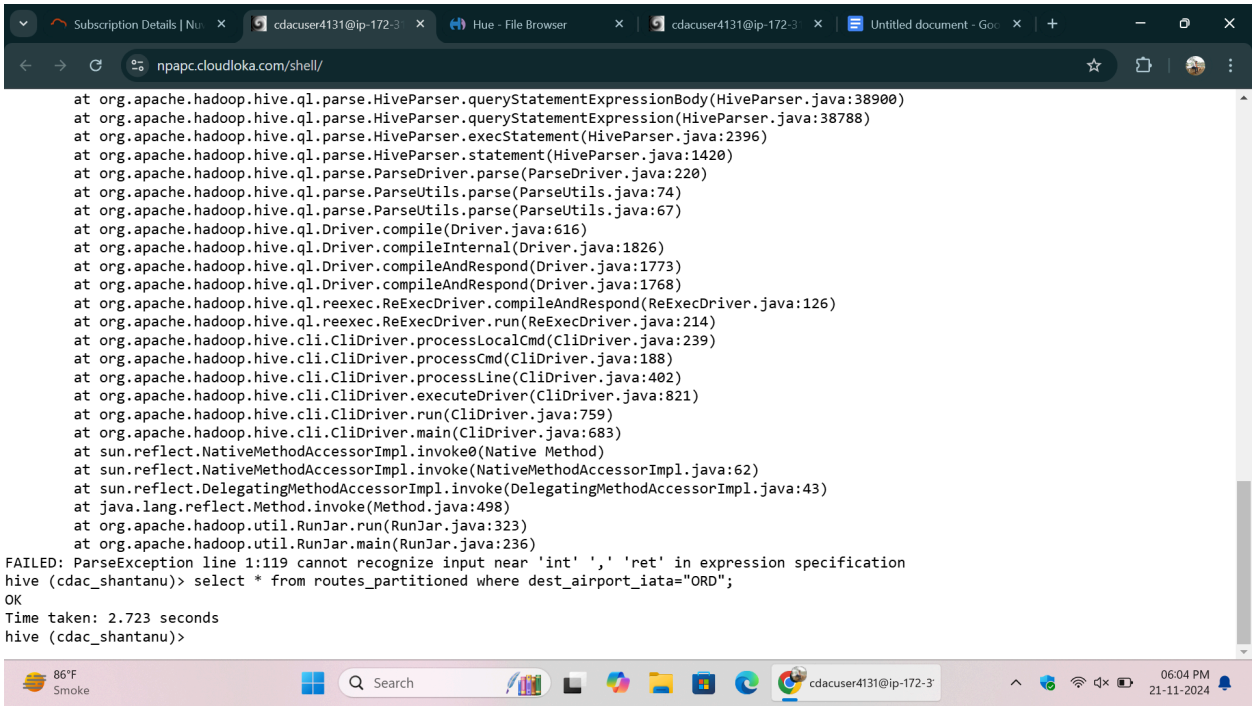
Insert value:-

```
        at org.apache.hadoop.hive.ql.parse.ParseDriver.parse(ParseDriver.java:220)
        at org.apache.hadoop.hive.ql.parse.ParseUtils.parse(ParseUtils.java:74)
        at org.apache.hadoop.hive.ql.parse.ParseUtils.parse(ParseUtils.java:67)
        at org.apache.hadoop.hive.ql.Driver.compile(Driver.java:616)
        at org.apache.hadoop.hive.ql.Driver.compileInternal(Driver.java:1826)
        at org.apache.hadoop.hive.ql.Driver.compileAndRespond(Driver.java:1773)
        at org.apache.hadoop.hive.ql.Driver.compileAndRespond(Driver.java:1768)
        at org.apache.hadoop.hive.ql.reexec.ReExecDriver.compileAndRespond(ReExecDriver.java:126)
        at org.apache.hadoop.hive.ql.reexec.ReExecDriver.run(ReExecDriver.java:214)
        at org.apache.hadoop.hive.cli.CliDriver.processLocalCmd(CliDriver.java:239)
        at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:188)
        at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:402)
        at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:821)
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:759)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:683)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:323)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:236)
FAILED: ParseException line 1:193 mismatched input 'src_airport_iata' expecting ) near 'int' in create table statement
hive (cdac_shantanu)> create table routes_partitioned( airline_iata string, airline_id int, src_airport_iata string, src_airport_id int, des
t_airport_id int, codeshare string, stops int, equipment string) partitioned by(dest_airport_iata string) row format delimited fields termin
ated by ',' stored as textfile;
OK
Time taken: 0.072 seconds
hive (cdac_shantanu)> insert overwrite table routes_partitioned partition(dest_airport_iata) select r.airline_iata string, r.airline_id int,
 r.src_airport_iata string, r.src_airport_id int,r.dest_airport_iata string, r.dest_airport_id int, r.codeshare string, r.stops int, r.equip
ment string from routes r distributed by dest_airport_iata;
```

2)insert overwrite table routes_partitioned partition(dest_airport_iata) select r.airline_iata string, r.airline_id int,r.src_airport_iata string, r.src_airport_id int,r.dest_airport_iata string, r.dest_airport_id int, r.codeshare string, r.stops int, r.equipment string from routes r distributed by dest_airport_iata;



3) select * from routes_partitioned where dest_airport_iata="ORD";



```
        at org.apache.hadoop.hive.ql.parse.HiveParser.queryStatementExpressionBody(HiveParser.java:38900)
        at org.apache.hadoop.hive.ql.parse.HiveParser.queryStatementExpression(HiveParser.java:38788)
        at org.apache.hadoop.hive.ql.parse.HiveParser.execStatement(HiveParser.java:2396)
        at org.apache.hadoop.hive.ql.parse.HiveParser.statement(HiveParser.java:1420)
        at org.apache.hadoop.hive.ql.parse.ParseDriver.parse(ParseDriver.java:220)
        at org.apache.hadoop.hive.ql.parse.ParseUtils.parse(ParseUtils.java:74)
        at org.apache.hadoop.hive.ql.parse.ParseUtils.parse(ParseUtils.java:67)
        at org.apache.hadoop.hive.ql.Driver.compile(Driver.java:616)
        at org.apache.hadoop.hive.ql.Driver.compileInternal(Driver.java:1826)
        at org.apache.hadoop.hive.ql.Driver.compileAndRespond(Driver.java:1773)
        at org.apache.hadoop.hive.ql.Driver.compileAndRespond(Driver.java:1768)
        at org.apache.hadoop.hive.ql.reexec.ReExecDriver.compileAndRespond(ReExecDriver.java:126)
        at org.apache.hadoop.hive.ql.reexec.ReExecDriver.run(ReExecDriver.java:214)
        at org.apache.hadoop.hive.cli.CliDriver.processLocalCmd(CliDriver.java:239)
        at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:188)
        at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:402)
        at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:821)
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:759)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:683)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:323)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:236)
FAILED: ParseException line 1:119 cannot recognize input near 'int' ',' 'ret' in expression specification
hive (cdac_shantanu)> select * from routes_partitioned where dest_airport_iata="ORD";
OK
Time taken: 2.723 seconds
hive (cdac_shantanu)>
```

II . Spark

Q.1)

1) booked_seat = air_split.map(lambda x : x[3] >= 20000 and x[3] <= 50000)
   For count :- booked_seat.count()

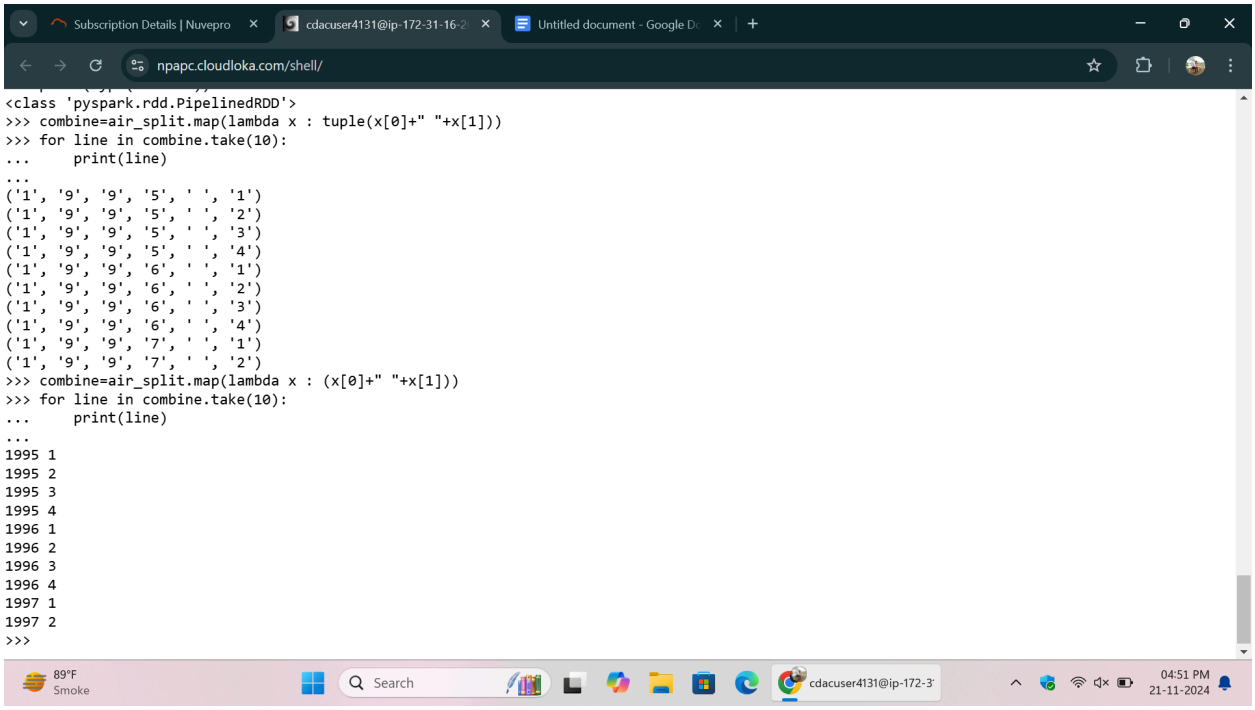
```
True
True
True
True
True
True
True
True
>>> combine.collect()
[True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, T
rue, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, Tru
e, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True, True, True, True, True, True]
>>> combine = air_split.map(lambda x: (x[3] >= 20000 and x[3] <=50000))
>>> combine.collect()
[True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, T
rue, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, Tru
e, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True, True, True, True, True, True]
>>> combine.count()
84
>>> combine = air_split.map(lambda x: x[3] >= 20000 and x[3] <=50000)
>>> combine.count()
84
>>> airline.count()
85
>>> booked_seat = air_split.map(lambda x: x[3] >= 20000 and x[3] <=50000)
>>> booked_seat.count()
84
>>>
```

2) combine=air_split.map(lambda x : (x[0]+" "+x[1]))

```
for line in combine.take(10):
        print(line)
```

In tuple:-

```
combine=air_split.map(lambda x : tuple(x[0]+" "+x[1]))
    for line in combine.take(10):
        print(line)
```



```
<class 'pyspark.rdd.PipelinedRDD'>
>>> combine=air_split.map(lambda x : tuple(x[0]+" "+x[1]))
>>> for line in combine.take(10):
...     print(line)
...
('1', '9', '9', '5', ' ', '1')
('1', '9', '9', '5', ' ', '2')
('1', '9', '9', '5', ' ', '3')
('1', '9', '9', '5', ' ', '4')
('1', '9', '9', '6', ' ', '1')
('1', '9', '9', '6', ' ', '2')
('1', '9', '9', '6', ' ', '3')
('1', '9', '9', '6', ' ', '4')
('1', '9', '9', '7', ' ', '1')
('1', '9', '9', '7', ' ', '2')
>>> combine=air_split.map(lambda x : (x[0]+" "+x[1]))
>>> for line in combine.take(10):
...     print(line)
...
1995 1
1995 2
1995 3
1995 4
1996 1
1996 2
1996 3
1996 4
1997 1
1997 2
>>>
```

Q.2)

minimum_seat=air_split.min(key= lambda x: x[3])
print(minimum_seat)
Output :- ('2000', '4', 340.08, 30103)

maximum_seat = air_split.max(key = lambda X:X[3])
print(maximum_seat)
Output :- ('2010', '1', 328.12, 49678)

average_seat = air_split.map(lambda x: x[3]).mean()
print(average_seat)
39640.70238095238

```
1995 1
1995 2
1995 3
1995 4
1996 1
1996 2
1996 3
1996 4
1997 1
1997 2
>>> minimum_seat=air_split.min(key= lambda x: x[3])
>>> print(minimum_seat)
('2000', '4', 340.08, 30103)
>>> maximum_seat = air_split.max(key = lambda X:X[3])
>>> print(maximum_seat)
('2010', '1', 328.12, 49678)
>>> average_seat = air_split.map(lambda x: x[3]).mean()
>>> print(average_seat)
39640.70238095238
>>> print(round(average_seat,2))
39640.7
>>> count_seat=air_split.count(key = lambda X:X[3])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: count() got an unexpected keyword argument 'key'
>>> count_seat=air_split[3].count()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'PipelinedRDD' object is not subscriptable
>>>
```

2) count_row=air_split.map(lambda x: (x[2] > 290.00))

>>> count_row.count()

```
('2000', '4', 340.08, 30103)
>>> maximum_seat = air_split.max(key = lambda X:X[3])
>>> print(maximum_seat)
('2010', '1', 328.12, 49678)
>>> average_seat = air_split.map(lambda x: x[3]).mean()
>>> print(average_seat)
39640.70238095238
>>> print(round(average_seat,2))
39640.7
>>> count_seat=air_split.count(key = lambda X:X[3])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: count() got an unexpected keyword argument 'key'
>>> count_seat=air_split[3].count()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'PipelinedRDD' object is not subscriptable
>>> count_row=air_split.map(lambda x: x[2] <290.00)
>>> count_row.count()
84
>>> count_row=air_split.map(lambda x: x[2] < 290.00)
>>> count_row.count()
84
>>> count_row=air_split.map(lambda x: x[2] > 290.00)
>>> count_row.count()
84
>>> count_row=air_split.map(lambda x: (x[2] > 290.00))
>>> count_row.count()
84
>>>
```

3) Combine = air_split.map(lambda x: x[1] , x[3]).mean

5) total_rev= air_split.map(lambda a: (a[2],a[2]*a[3]))

>>> total_rev.take(10)

[(296.9, 13823960.899999999), (296.8, 11113082.4), (287.51, 9812141.28), (287.78, 8745058.639999999), (283.97, 13576037.760000002), (275.78, 11864055.6), (269.49, 10497174.48), (278.33, 10421510.19), (283.4, 9937987.799999999), (289.44, 13477773.6)]

for i in total_rev.take(10):

... print(i)