

# BIG DATA HADOOP & SPARK TRAINING

Assignment on Spark MLIB I

Rashmi Krishna

## Aviation data analysis

The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS) tracks the

on-time performance of domestic flights operated by large air carriers. Summary information on the

number of on-time, delayed, canceled, and diverted flights appears in DOT's monthly Air Travel

Consumer Report, published about 30 days after the month's end, as well as in summary tables posted

on this website. Summary statistics and raw data are made available to the public at the time the Air

Travel Consumer Report is released.

You can download the datasets from the following links:

[https://drive.google.com/file/d/0B\\_Qjau8wv1KoWTVDUVFOdzlJNWM/view](https://drive.google.com/file/d/0B_Qjau8wv1KoWTVDUVFOdzlJNWM/view)

Screen shot of the input data set:

DelayedFlights - Excel

File Home Insert Page Layout Formulas Data Review View Tell me what you want to do...

Clipboard Font Alignment Number Styles Cells Editing

Input data set

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	sno	Year	Month	DayofMor	DayOfWe	DepTime	CRSDepTi	ArrTime	CRSArrTin	FlightNum	ActualElar	CRSElapse	AirTime	ArrDelay	DepDelay	Origin	Dest	Distance	TaxiIn	TaxiOut	Canceled
2	0	2008	1	3	4	2003	1955	2211	2225	335	128	150	116	-14	8	IAD	TPA	810	4	8	
3	1	2008	1	3	4	754	735	1002	1000	3231	128	145	113	2	19	IAD	TPA	810	5	10	
4	2	2008	1	3	4	628	620	804	750	448	96	90	76	14	8	IND	BWI	515	3	17	
5	4	2008	1	3	4	1829	1755	1959	1925	3920	90	90	77	34	34	IND	BWI	515	3	10	
6	5	2008	1	3	4	1940	1915	2121	2110	378	101	115	87	11	25	IND	JAX	688	4	10	
7	6	2008	1	3	4	1937	1830	2037	1940	509	240	250	230	57	67	IND	LAS	1591	3	7	
8	10	2008	1	3	4	706	700	916	915	100	130	135	106	1	6	IND	MCO	828	5	19	
9	11	2008	1	3	4	1644	1510	1845	1725	1333	121	135	107	80	94	IND	MCO	828	6	8	
10	15	2008	1	3	4	1029	1020	1021	1010	2272	52	50	37	11	9	IND	MDW	162	6	9	
11	16	2008	1	3	4	1452	1425	1640	1625	675	228	240	213	15	27	IND	PHX	1489	7	8	
12	17	2008	1	3	4	754	745	940	955	1144	226	250	205	-15	9	IND	PHX	1489	5	16	
13	18	2008	1	3	4	1323	1255	1526	1510	4	123	135	110	16	28	IND	TPA	838	4	9	
14	19	2008	1	3	4	1416	1325	1512	1435	54	56	70	49	37	51	ISP	BWI	220	2	5	
15	21	2008	1	3	4	1657	1625	1754	1735	623	57	70	47	19	32	ISP	BWI	220	5	5	
16	22	2008	1	3	4	1900	1840	1956	1950	717	56	70	49	6	20	ISP	BWI	220	2	5	
17	23	2008	1	3	4	1039	1030	1133	1140	1244	54	70	47	-7	9	ISP	BWI	220	2	5	
18	25	2008	1	3	4	1520	1455	1619	1605	2553	59	70	50	14	25	ISP	BWI	220	2	7	
19	26	2008	1	3	4	1422	1255	1657	1610	188	155	195	143	47	87	ISP	FLL	1093	6	6	
20	27	2008	1	3	4	1954	1925	2239	2235	1754	165	190	155	4	29	ISP	FLL	1093	3	7	
21	30	2008	1	3	4	2107	1945	2334	2230	362	147	165	134	64	82	ISP	MCO	972	6	7	
22	33	2008	1	3	4	1312	1300	1546	1550	1397	154	170	140	-4	12	ISP	MCO	972	7	7	
23	34	2008	1	3	4	1449	1430	1715	1720	3398	146	170	134	-5	19	ISP	MCO	972	6	6	

DelayedFlights

## Delayed\_Flights.csv Datasets

There are 29 columns in this dataset. Some of them have been mentioned below:

- Year: 1987 – 2008
- Month: 1 – 12
- FlightNum: Flight number
- Canceled: Was the flight canceled?
- CancellationCode: The reason for cancellation.

For complete details, refer to this link.

### Program to accomplish the below tasks:

This part of the program reads the CSV file as per the schema defined, convert to dataframe and register as temporary table called “**Flights\_Table**”

```
package MLIB

import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions._
import org.apache.spark.sql.types.{LongType, StringType, StructField, StructType}

object asnt27 {

  def main(args: Array[String]): Unit = {

    // create a spark session object

    val spark = SparkSession
      .builder()
      .master("local")
      .appName("MLIB example")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()
    println("Spark Session Object created")

    // create a variable "Flight" to save the contents of the csv file and convert to dataframe
    and register a temporary table on the data frame
    val Flight = spark.read.format("CSV").option("header", true).load("E:\\assignments
ss\\DelayedFlights.csv")
    val Fl = Flight.toDF()
    Flight.show()
    Fl.registerTempTable("Flights_Table")
    println("Flights_Table is registered!")
  }
}
```

```
NewProj1 [C:\Users\Laptop\IdeaProjects\NewProj1] - ...src\main\scala\MLIB\asnt27.scala [newproj1] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

NewProj1 | src | main | scala | MLIB | asnt27.scala

Run: asnt27 x
"C:\Program Files\Java\jdk1.8.0_161\bin\java.exe" ...
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
18/06/24 11:35:54 INFO SparkContext: Running Spark version 2.1.0
18/06/24 11:35:55 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/06/24 11:35:55 INFO SecurityManager: Changing view acls to: Laptop
18/06/24 11:35:55 INFO SecurityManager: Changing modify acls to: Laptop
18/06/24 11:35:55 INFO SecurityManager: Changing view acls groups to:
18/06/24 11:35:55 INFO SecurityManager: Changing modify acls groups to:
18/06/24 11:35:55 INFO SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(Laptop); groups with view permissions: Set(); user
18/06/24 11:35:56 INFO Utils: Successfully started service 'sparkDriver' on port 62204.
18/06/24 11:35:56 INFO SparkEnv: Registering MapOutputTracker
18/06/24 11:35:56 INFO SparkEnv: Registering BlockManagerMaster
18/06/24 11:35:56 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
18/06/24 11:35:56 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
18/06/24 11:35:56 INFO DiskBlockManager: Created local directory at C:\Users\Laptop\AppData\Local\Temp\blockmgr-72911522-6c4b-4547-baec-913b665dd609
18/06/24 11:35:56 INFO MemoryStore: MemoryStore started with capacity 902.7 MB
18/06/24 11:35:56 INFO SparkEnv: Registering OutputCommitCoordinator
18/06/24 11:35:57 INFO Utils: Successfully started service 'SparkUI' on port 4040.
18/06/24 11:35:57 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://192.168.56.1:4040
18/06/24 11:35:57 INFO Executor: Starting executor ID driver on host localhost
18/06/24 11:35:57 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 62213.
18/06/24 11:35:57 INFO NettyBlockTransferService: Server created on 192.168.56.1:62213
18/06/24 11:35:57 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
18/06/24 11:35:57 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, 192.168.56.1, 62213, None)
18/06/24 11:35:57 INFO BlockManagerMasterEndpoint: Registering block manager 192.168.56.1:62213 with 902.7 MB RAM, BlockManagerId(driver, 192.168.56.1, 62213, None)
18/06/24 11:35:57 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, 192.168.56.1, 62213, None)
18/06/24 11:35:57 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, 192.168.56.1, 62213, None)
18/06/24 11:35:57 INFO SharedState: Warehouse path is 'file:/C:/Users/Laptop/IdeaProjects/NewProj1/spark-warehouse/'.
Spark Session Object created
18/06/24 11:35:59 INFO MemoryStore: Block broadcast_0 stored as values in memory (estimated size 127.1 KB, free 902.6 MB)
18/06/24 11:35:59 INFO MemoryStore: Block broadcast_0_piece0 stored as bytes in memory (estimated size 14.3 KB, free 902.6 MB)
18/06/24 11:35:59 INFO BlockManagerInfo: Added broadcast_0_piece0 in memory on 192.168.56.1:62213 (size: 14.3 KB, free: 902.7 MB)
18/06/24 11:35:59 INFO SparkContext: Created broadcast 0 from load at asnt27.scala:49
18/06/24 11:35:59 ERROR Shell: Failed to locate the winutils binary in the hadoop binary path
```

```
NewProj1 [C:\Users\Laptop\IdeaProjects\NewProj1] - ...src\main\scala\MLIB\asnt27.scala [newproj1] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

NewProj1 | src | main | scala | MLIB | asnt27.scala

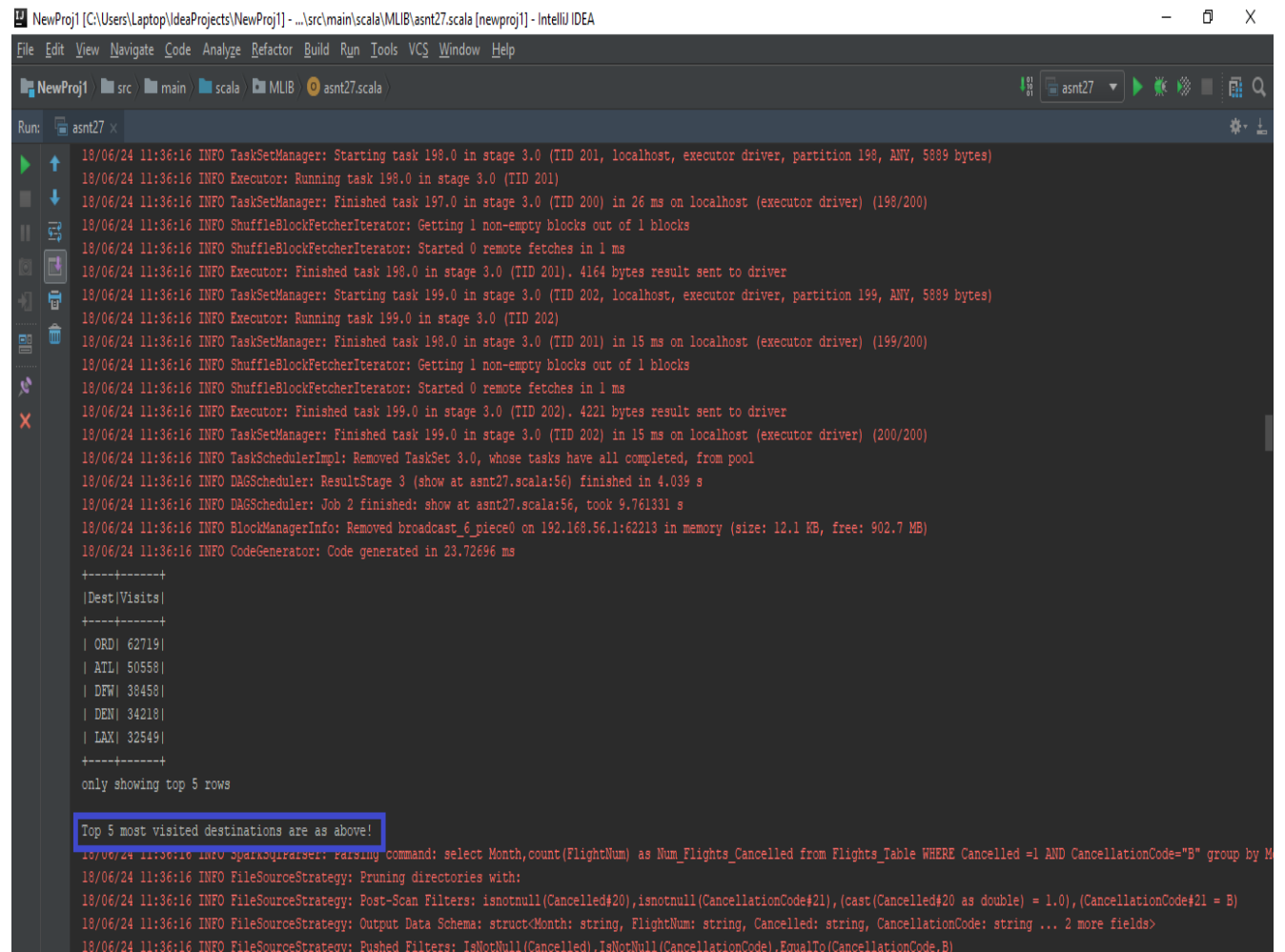
Run: asnt27 x
18/06/24 11:36:00 INFO DAGScheduler: Missing parents: List()
18/06/24 11:36:00 INFO DAGScheduler: Submitting ResultStage 1 (MapPartitionsRDD[5] at load at asnt27.scala:49), which has no missing parents
18/06/24 11:36:00 INFO MemoryStore: Block broadcast_3 stored as values in memory (estimated size 3.5 KB, free 902.4 MB)
18/06/24 11:36:00 INFO MemoryStore: Block broadcast_3_piece0 stored as bytes in memory (estimated size 2.1 KB, free 902.4 MB)
18/06/24 11:36:00 INFO BlockManagerInfo: Added broadcast_3_piece0 in memory on 192.168.56.1:62213 (size: 2.1 KB, free: 902.7 MB)
18/06/24 11:36:00 INFO SparkContext: Created broadcast 3 from broadcast at DAGScheduler.scala:996
18/06/24 11:36:00 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 1 (MapPartitionsRDD[5] at load at asnt27.scala:49)
18/06/24 11:36:00 INFO TaskSchedulerImpl: Adding task set 1.0 with 1 tasks
18/06/24 11:36:00 INFO TaskSetManager: Starting task 0.0 in stage 1.0 (TID 1, localhost, executor driver, partition 0, PROCESS_LOCAL, 5988 bytes)
18/06/24 11:36:00 INFO Executor: Running task 0.0 in stage 1.0 (TID 1)
18/06/24 11:36:00 INFO HadoopRDD: Input split: file:/E:/assignments/ss/DelayedFlights.csv?+33554432
18/06/24 11:36:00 INFO Executor: Finished task 0.0 in stage 1.0 (TID 1). 1283 bytes result sent to driver
18/06/24 11:36:00 INFO TaskSetManager: Finished task 0.0 in stage 1.0 (TID 1) in 48 ms on localhost (executor driver) (1/1)
18/06/24 11:36:00 INFO TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
18/06/24 11:36:00 INFO DAGScheduler: ResultStage 1 (load at asnt27.scala:49) finished in 0.051 s
18/06/24 11:36:00 INFO DAGScheduler: Job 1 finished: load at asnt27.scala:49, took 0.096066 s
18/06/24 11:36:01 INFO BlockManagerInfo: Removed broadcast_1_piece0 on 192.168.56.1:62213 in memory (size: 2.1 KB, free: 902.7 MB)
18/06/24 11:36:01 INFO BlockManagerInfo: Removed broadcast_2_piece0 on 192.168.56.1:62213 in memory (size: 14.3 KB, free: 902.7 MB)
18/06/24 11:36:01 INFO BlockManagerInfo: Removed broadcast_3_piece0 on 192.168.56.1:62213 in memory (size: 2.1 KB, free: 902.7 MB)
18/06/24 11:36:04 INFO SparkSqlParser: Parsing command: Flights_Table
Flights_Table is registered!
18/06/24 11:36:04 INFO SparkSqlParser: Parsing command: select Dest,count(dest) as Visits from Flights_Table group by Dest
18/06/24 11:36:05 INFO FileSourceStrategy: Pruning directories with:
18/06/24 11:36:05 INFO FileSourceStrategy: Post-Scan Filters:
18/06/24 11:36:05 INFO FileSourceStrategy: Output Data Schema: struct<Dest: string>
18/06/24 11:36:05 INFO FileSourceStrategy: Pushed Filters:
18/06/24 11:36:05 WARN Utils: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.debug.maxToStringFields' in SparkEnv
```

## Problem Statement 1: Find out the top 5 most visited destinations.

- Create a variable called `dest`, which holds the result of the query to find the top most visited destinations.
- Query: we select column “Dest” and count the number of Dest entries from the registered temporary table and group them by Dest column and order them in descending order.
- Filter them with top 5

//Find out the top 5 most visited destinations.

```
val dest = spark.sql("""select Dest,count(dest) as Visits from Flights_Table group by Dest
""").toDF()
dest.sort(desc("Visits")).show(5)
println("Top 5 most visited destinations are as above!")
```



NewProj1 [C:\Users\Laptop\IdeaProjects\NewProj1] - ... \src\main\scala\MLIB\asnt27.scala [newproj1] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

NewProj1 src main MLIB asnt27.scala

Run: asnt27 x

```
18/06/24 11:36:16 INFO TaskSetManager: Starting task 198.0 in stage 3.0 (TID 201, localhost, executor driver, partition 198, ANY, 5889 bytes)
18/06/24 11:36:16 INFO Executor: Running task 198.0 in stage 3.0 (TID 201)
18/06/24 11:36:16 INFO TaskSetManager: Finished task 197.0 in stage 3.0 (TID 200) in 26 ms on localhost (executor driver) (198/200)
18/06/24 11:36:16 INFO ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
18/06/24 11:36:16 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 1 ms
18/06/24 11:36:16 INFO Executor: Finished task 198.0 in stage 3.0 (TID 201). 4164 bytes result sent to driver
18/06/24 11:36:16 INFO TaskSetManager: Starting task 199.0 in stage 3.0 (TID 202, localhost, executor driver, partition 199, ANY, 5889 bytes)
18/06/24 11:36:16 INFO Executor: Running task 199.0 in stage 3.0 (TID 202)
18/06/24 11:36:16 INFO TaskSetManager: Finished task 198.0 in stage 3.0 (TID 201) in 15 ms on localhost (executor driver) (199/200)
18/06/24 11:36:16 INFO ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
18/06/24 11:36:16 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 1 ms
18/06/24 11:36:16 INFO Executor: Finished task 199.0 in stage 3.0 (TID 202). 4221 bytes result sent to driver
18/06/24 11:36:16 INFO TaskSetManager: Finished task 199.0 in stage 3.0 (TID 202) in 15 ms on localhost (executor driver) (200/200)
18/06/24 11:36:16 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
18/06/24 11:36:16 INFO DAGScheduler: ResultStage 3 (show at asnt27.scala:56) finished in 4.039 s
18/06/24 11:36:16 INFO DAGScheduler: Job 2 finished: show at asnt27.scala:56, took 9.761331 s
18/06/24 11:36:16 INFO BlockManagerInfo: Removed broadcast_6_piece0 on 192.168.56.1:62213 in memory (size: 12.1 KB, free: 902.7 MB)
18/06/24 11:36:16 INFO CodeGenerator: Code generated in 23.72696 ms

+-----+
|Dest|Visits|
+-----+
| ORD| 62719|
| ATL| 50558|
| DFW| 38458|
| DEN| 34218|
| LAX| 32549|
+-----+

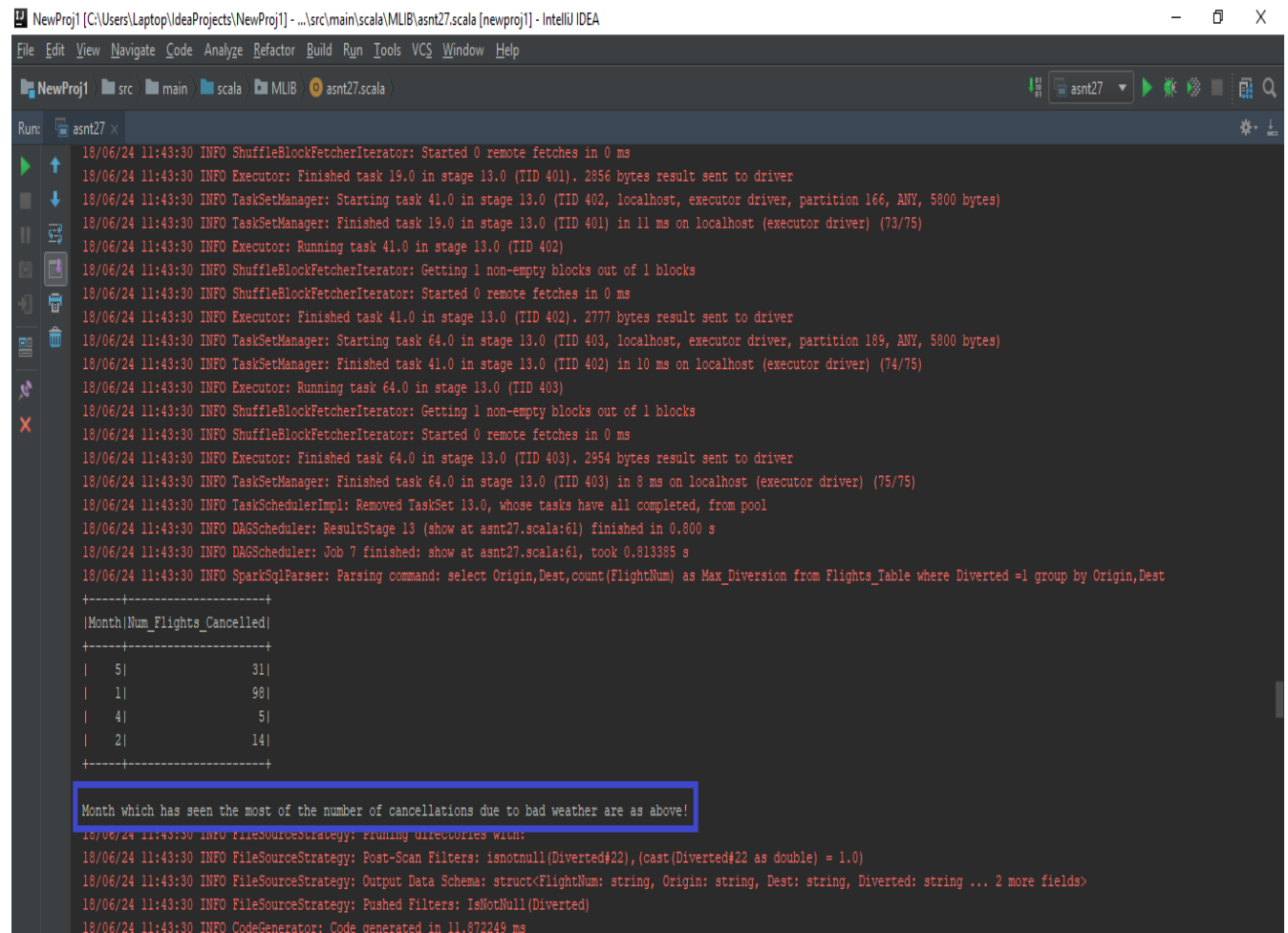
only showing top 5 rows

Top 5 most visited destinations are as above!
18/06/24 11:36:16 INFO SparkQueryRunner: Parsing command: select Month,count(FlightNum) as Num_Flights_Cancelled from Flights_Table WHERE Cancelled =1 AND CancellationCode="B" group by M
18/06/24 11:36:16 INFO FileSourceStrategy: Pruning directories with:
18/06/24 11:36:16 INFO FileSourceStrategy: Post-Scan Filters: isNotNull(Cancelled#20),isNotNull(CancellationCode#21),(cast(Cancelled#20 as double) = 1.0),(CancellationCode#21 = B)
18/06/24 11:36:16 INFO FileSourceStrategy: Output Data Schema: struct<Month: string, FlightNum: string, Cancelled: string, CancellationCode: string ... 2 more fields>
18/06/24 11:36:16 INFO FileSourceStrategy: Pushed Filters: IsNotNull(Cancelled),IsNotNull(CancellationCode),EqualTo(CancellationCode,B)
```

## Problem Statement 2: Which month has seen the most number of cancellations due to bad weather?

- Create a variable called “cancel”, which holds the result of the query to find which month has seen most number of cancellations due to bad weather.
- Query: we select column Month and count the number of FlightNumber and name it as Num\_Flights\_Cancelled from the registered table.
- Filter them where column “cancelled” is 1 and cancellation code is “B”, which indicates the number of flights cancelled due to bad weather.

```
//Which month has seen the most number of cancellations due to bad weather?  
val cancel = spark.sql("""select Month,count(FlightNum) as Num_Flights_Cancelled from  
Flights_Table WHERE Cancelled =1 AND CancellationCode="B" group by Month """).toDF()  
cancel.show()  
println("Month which has seen the most of the number of cancellations due to bad weather  
are as above!")
```

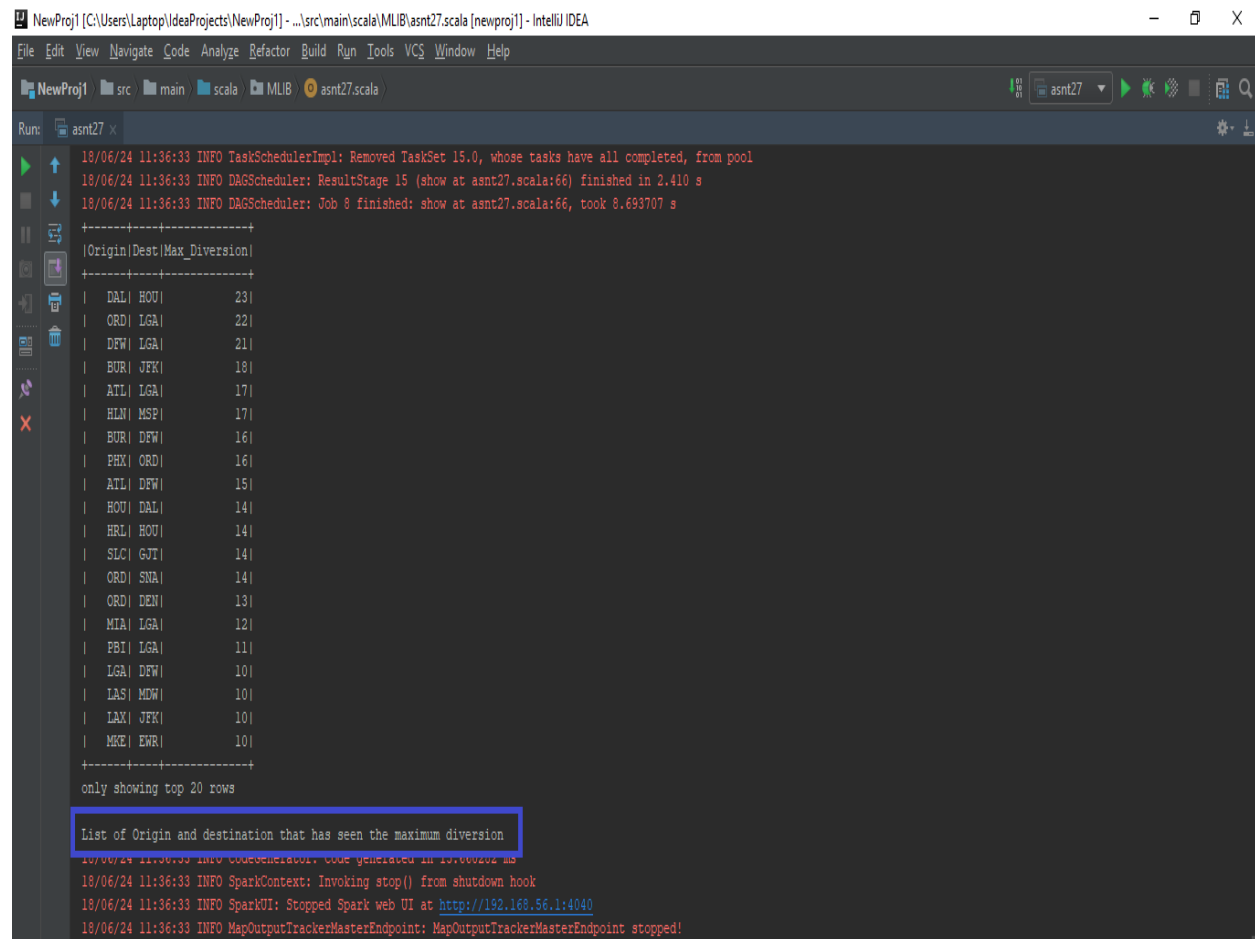


```
NewProj1 [C:\Users\Laptop\IdeaProjects\NewProj1] - ...src\main\scala\MLIB\asnt27.scala [newproj1] - IntelliJ IDEA  
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help  
NewProj1 src main scala MLIB asnt27.scala  
Run: asnt27 x  
18/06/24 11:43:30 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms  
18/06/24 11:43:30 INFO Executor: Finished task 19.0 in stage 13.0 (TID 401). 2856 bytes result sent to driver  
18/06/24 11:43:30 INFO TaskSetManager: Starting task 41.0 in stage 13.0 (TID 402, localhost, executor driver, partition 166, ANY, 5800 bytes)  
18/06/24 11:43:30 INFO TaskSetManager: Finished task 19.0 in stage 13.0 (TID 401) in 11 ms on localhost (executor driver) (73/75)  
18/06/24 11:43:30 INFO Executor: Running task 41.0 in stage 13.0 (TID 402)  
18/06/24 11:43:30 INFO ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks  
18/06/24 11:43:30 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms  
18/06/24 11:43:30 INFO Executor: Finished task 41.0 in stage 13.0 (TID 402). 2777 bytes result sent to driver  
18/06/24 11:43:30 INFO TaskSetManager: Starting task 64.0 in stage 13.0 (TID 403, localhost, executor driver, partition 189, ANY, 5800 bytes)  
18/06/24 11:43:30 INFO TaskSetManager: Finished task 41.0 in stage 13.0 (TID 402) in 10 ms on localhost (executor driver) (74/75)  
18/06/24 11:43:30 INFO Executor: Running task 64.0 in stage 13.0 (TID 403)  
18/06/24 11:43:30 INFO ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks  
18/06/24 11:43:30 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms  
18/06/24 11:43:30 INFO Executor: Finished task 64.0 in stage 13.0 (TID 403). 2954 bytes result sent to driver  
18/06/24 11:43:30 INFO TaskSetManager: Finished task 64.0 in stage 13.0 (TID 403) in 8 ms on localhost (executor driver) (75/75)  
18/06/24 11:43:30 INFO TaskSchedulerImpl: Removed TaskSet 13.0, whose tasks have all completed, from pool  
18/06/24 11:43:30 INFO DAGScheduler: ResultStage 13 (show at asnt27.scala:61) finished in 0.800 s  
18/06/24 11:43:30 INFO DAGScheduler: Job 7 finished: show at asnt27.scala:61, took 0.813385 s  
18/06/24 11:43:30 INFO SparkSqlParser: Parsing command: select Origin, Dest, count(FlightNum) as Max_Diversion from Flights_Table where Diverted =1 group by Origin, Dest  
+-----+  
|Month|Num_Flights_Cancelled|  
+-----+  
| 5|31|  
| 1|98|  
| 4|5|  
| 2|14|  
+-----+  
Month which has seen the most of the number of cancellations due to bad weather are as above!  
18/06/24 11:43:30 INFO FileSourceStrategy: Pruning directories with:  
18/06/24 11:43:30 INFO FileSourceStrategy: Post-Scan Filters: isNotNull(Diverted#22), (cast(Diverted#22 as double) = 1.0)  
18/06/24 11:43:30 INFO FileSourceStrategy: Output Data Schema: struct<FlightNum: string, Origin: string, Dest: string, Diverted: string ... 2 more fields>  
18/06/24 11:43:30 INFO FileSourceStrategy: Pushed Filters: IsNotNull(Diverted)  
18/06/24 11:43:30 INFO CodeGenerator: Code generated in 11.872249 ms
```

## Problem Statement 3: Which route (origin & destination) has seen the maximum diversion?

- Create a variable called “diversion”, which holds the result of the query to find origin and destination which has seen maximum diversion.
- Query: we select columns Origin, Dest and count the number of Flights and name it as Max\_Diversion from the registered table.
- Sort the column “Max\_Diversion” in descending order.

```
//Which route (origin & destination) has seen the maximum diversion?  
val diversion = spark.sql("""select Origin, Dest, count(FlightNum) as Max_Diversion from  
Flights_Table where Diverted =1 group by Origin, Dest """)  
diversion.toDF().sort(desc("Max_Diversion")).show()  
println("List of Origin and destination that has seen the maximum diversion")
```



The screenshot shows the IntelliJ IDEA interface with a Scala file named `asnt27.scala` open. The Run console at the bottom displays the execution of the Spark SQL query. The output shows the top 20 routes by maximum diversion, sorted in descending order. The route `DAL|HOU` has the highest diversion count of 23.

Origin	Dest	Max_Diversion
DAL	HOU	23
ORD	LGA	22
DFW	LGA	21
BUR	JFK	18
ATL	LGA	17
HLN	MSP	17
BUR	DFW	16
PHX	ORD	16
ATL	DFW	15
HOU	DAL	14
HRL	HOU	14
SLC	GJT	14
ORD	SNA	14
ORD	DFW	13
MIA	LGA	12
FBI	LGA	11
LGA	DFW	10
LAS	MDW	10
LAX	JFK	10
MKE	ENR	10

only showing top 20 rows

List of Origin and destination that has seen the maximum diversion