Session 21 : SPARK

Assignment 2

**Problem Statement**

Implement the below blog at your end and send the complete documentation.

<https://drive.google.com/file/d/0B_Qjau8wv1KoUThzZ24tT1NsZGs/view?usp=sharing>

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

Delayed\_Flights.csv

**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?

• CancelleationCode: The reason for cancellation.

For complete details, refer to this link.

**Problem Statement 1**

**Find out the top 5 most visited destinations.**

**Problem Statement 2**

**Which month has seen the most number of cancellations due to bad weather?**

**Problem Statement 3**

**Top ten origins with the highest AVG departure delay**

**Problem Statement 4**

**Which route (origin & destination) has seen the maximum diversion?**

**(Solution from next page...............)**

**Solution:-**

Starting Spark-shell

Command – spark-shell



**Problem Statement 1**

**Find out the top 5 most visited destinations.**

**scala>** val delayed\_flights = sc.textFile("DelayedFlights.csv");

**scala>** val mapping = delayed\_flights.map(x => x.split(",")).map(x => (x(18),1)).filter(x =>x.\_1!=null).reduceByKey(\_+\_).map(x => (x.\_2,x.\_1)).sortByKey(false).map(x => (x.\_2,x.\_1)).take(5);



**Problem Statement 2**

**Which month has seen the most number of cancellations due to bad weather?**

**scala>** val delayed\_flights = sc.textFile("DelayedFlights.csv");

**scala>** val canceled = delayed\_flights.map(x => x.split(",")).filter(x => ((x(22).equals("1")) && (x(23).equals("B")))).map(x => (x(2),1)).reduceByKey(\_+\_).map(x =>(x.\_2,x.\_1)).sortByKey(false).map(x =>(x.\_2,x.\_1)).take(1);



**Problem Statement 3**

**Top ten origins with the highest AVG departure delay**

**scala>** val delayed\_flights = sc.textFile("DelayedFlightsNoHeader.csv");

**scala>** val avg = delayed\_flights.map(x => x.split(",")).map(x => (x(17),x(16).toDouble)).mapValues((\_,1)).reduceByKey((x, y) => (x.\_1 + y.\_1, x.\_2 + y.\_2)).mapValues{ case (sum, count) => (1.0 \*sum)/count}.map(x => (x.\_2,x.\_1)).sortByKey(false).map(x => (x.\_2,x.\_1)).take(10);



**Problem Statement 4**

**Which route (origin & destination) has seen the maximum diversion?**

**scala>** val delayed\_flights = sc.textFile("DelayedFlights.csv");

delayed\_flights: org.apache.spark.rdd.RDD[String] = DelayedFlights.csv MapPartitionsRDD[53] at textFile at <console>:24

**scala>** val diversion = delayed\_flights.map(x => x.split(",")).filter(x => ((x(24).equals("1")))).map(x =>((x(17)+","+x(18)),1)).reduceByKey(\_+\_).map(x => (x.\_2,x.\_1)).sortByKey(false).map(x =>(x.\_2,x.\_1)).take(10).foreach(println);

