**Session 21: MLIB AND GRAPHX**

**Assignment 21.2**

**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>

**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, cancelled, 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.

***Delayed\_Flights.csv*** Datasets

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

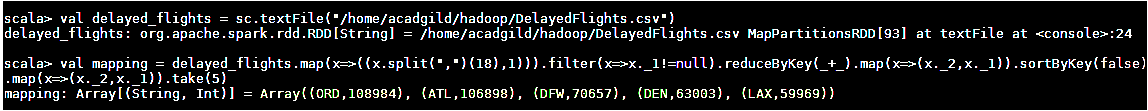
1. Year: 1987 – 2008
2. Month: 1 – 12
3. FlightNum: Flight number
4. Cancelled: Was the flight cancelled?
5. CancelleationCode: The reason for cancellation.

**Problem Statement 1 - Find out the top 5 most visited destinations.**

**Code**

***val delayed\_flights = sc.textFile("/home/acadgild/hadoop/DelayedFlights.csv")***

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



**Output**

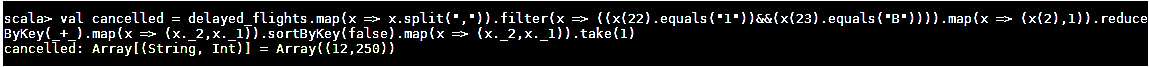
***mapping: Array[(String, Int)] = Array((ORD,108984), (ATL,106898), (DFW,70657), (DEN,63003), (LAX,59969))***

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

**Code**

***val cancelled = 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)***

**Output *cancelled: Array[(String, Int)] = Array((12,250))***

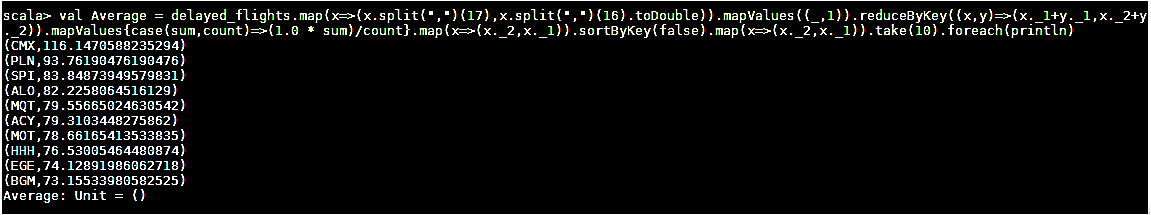


**Problem Statement 3 - Top ten origins with the highest AVG departure delay**

**Code**

**val Average = delayed\_flights.map(x=>(x.split(",")(17),x.split(",")(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)**

**Output**



**Problem Statement 4 - Which route (origin & destination) has seen the maximum diversion?**

**Code**

***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)***

**Output**

