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:

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.

Created a project in IDEA IntelliJ and added scala object.

Imported the required namespaces (jars)

Created spark object

```
//Let us create a spark session object
//Create a case class globally to be used inside the main method
val spark = SparkSession
  .builder()
  .master("local")
  .appName("Spark Machine Learning")
  .config("spark.some.config.option", "some-value")
  .getOrCreate()

//Set the log level as warning
spark.sparkContext.setLogLevel("WARN")
```

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

Solution Approach

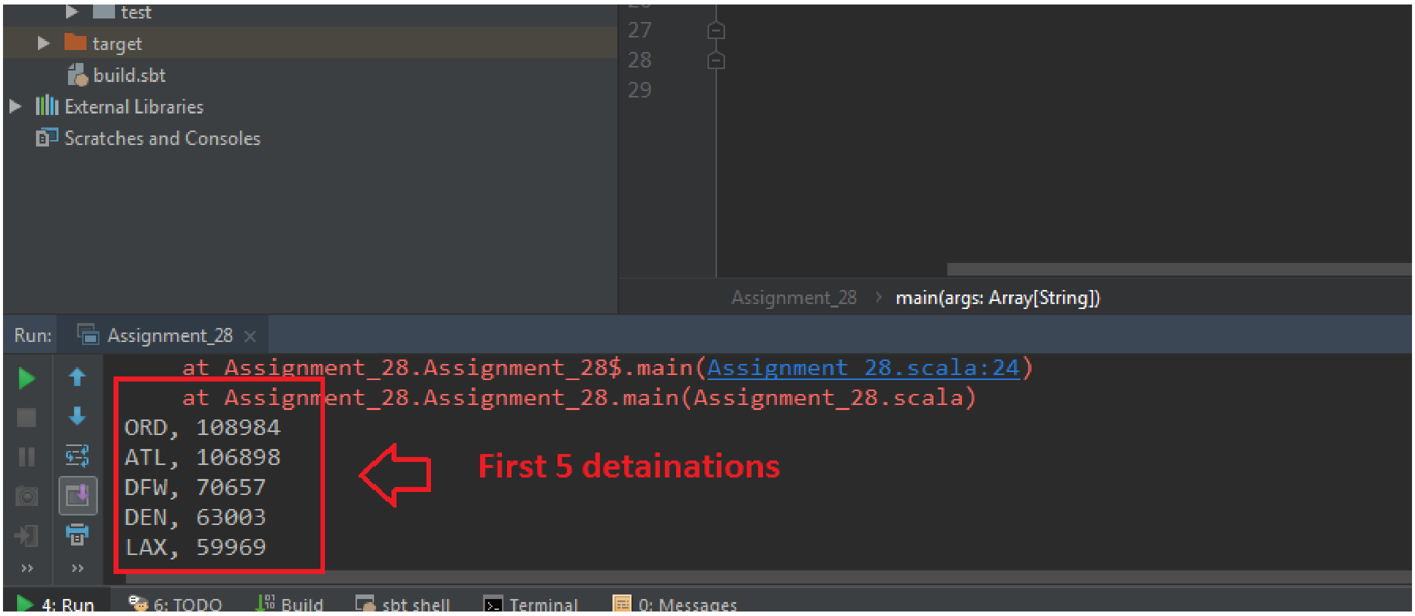
- Create the RDD using the textfile method of spark context
- Here we not are mapping the data to any case class schema instead using the schema from file only.
- 18th is the destination column, first split the data and get the value from 18th column and then map the same with counter 1.

- Get the sum of all counter based on key using **reducebykey** and sort the RDD using **sortByKey**
- **Take** the first 5

Source Code

```
val delayed_flights = spark.sparkContext.textFile("E:\\Prachi IMP\\Hadoop\\Day 
28\\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)
mapping.foreach(x=>println(x._1+", "+x._2))
```

Output



```
    at Assignment_28.Assignment_28$.main(Assignment_28.scala:24)
    at Assignment_28.Assignment_28.main(Assignment_28.scala)
ORD, 108984
ATL, 106898
DFW, 70657
DEN, 63003
LAX, 59969
```

First 5 detainations

Problem Statement 2

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

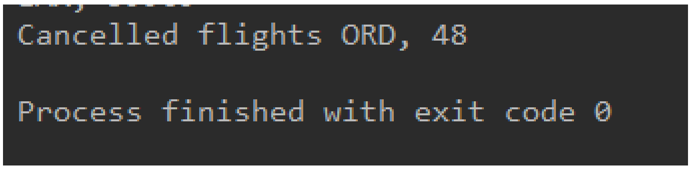
Source Code

```
//Task 2
val cancelled = delayed_flights.map(x => x.split(",")).filter(x => 
((x(22).equals("1"))&&(x(23).equals("B")))).map(x => 
(x(17),1)).reduceByKey(_+_).map(x => (x._2,x._1)).sortByKey(false).map(x => 
(x._2,x._1)).take(1)
cancelled.foreach(x=>println("Cancelled flights " + x._1+ ", " + x._2))
```

- Column 22 represents if flight has been cancelled

- Filter based on this column and map flight)key based) to counter to get the sum proceeded with maximum number of cancellations

Output

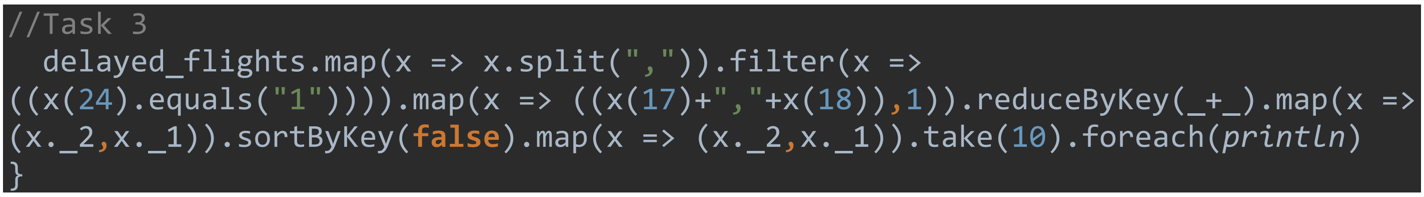


```
Cancelled flights ORD, 48

Process finished with exit code 0
```

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

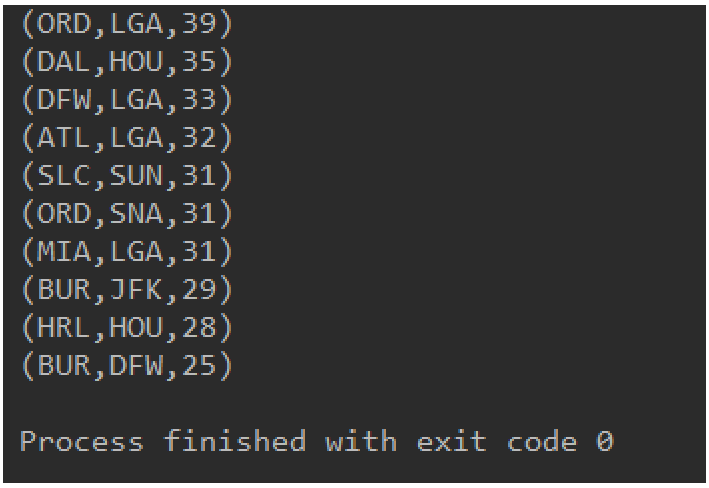
Source Code



```
//Task 3
  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)
}
```

- Column 24 represents if flight has been diverted or not
- Filtered on that column 24 and create a tuple with Origin (17) and Dest (18) and map it with a counter
- Get sum based on counter and order in descending order and take top 10.

Output



```
(ORD,LGA,39)
(DAL,HOU,35)
(DFW,LGA,33)
(ATL,LGA,32)
(SLC,SUN,31)
(ORD,SNA,31)
(MIA,LGA,31)
(BUR,JFK,29)
(HRL,HOU,28)
(BUR,DFW,25)

Process finished with exit code 0
```